# README file: Applying the m3index\_scoring and c3index\_scoring functions in R.

This set of notes is currently up to date for version 1.0 of the function. Please contact [james.stanley@otago.ac.nz](mailto:james.stanley@otago.ac.nz) for help or to report errors.

v1.0 was built and tested in R4.0.1 and uses packages called by tidyverse 1.3.0, with functions specific to the following tidyverse packages:

dplyr 1.0.8 stringr 1.4.0 tidyr 1.2.0

(package versions above are those used in testing rather than those necessary to run functions)

# Calculating the M3 index

The paper describing the development and validation of the M3 index (Stanley & Sarfati, 2017) is:

**Stanley J, Sarfati D. (2017) The new measuring multimorbidity index predicted mortality better than Charlson and Elixhauser indices among the general population. Journal of Clinical Epidemiology 2017;92:99-110.** **DOI: 10.1016/j.jclinepi.2017.08.005**

The sole function in the attached file is m3index\_scoring()– see the function file for a list of all options. The main options are demarcated below (with an example) for a typical usage scenario.

For the M3 index, the data is required to be in “long” format before processing (see example dataset). If you need to do this yourself, you can use the pivot\_longer() function from tidyr::

Example call to function for M3 index

m3index\_scoring <- function(in\_df=my\_diagnosis\_df,

clinical\_code\_col="diags",

id\_cols="patient\_id")

## Input Files:

*Overview*

The input dataframe (in\_df) should contain (at a minimum) identifying variable(s) that indicate individual people, and one or more columns with ICD-10 codes for diagnoses recorded for hospital admissions.

This file should be pre-prepared by the user so that it contains only those diagnoses that fall within the lookback eligibility period (for M3 and C3, we consider this as the five year period up to and including the index hospitalisation/diagnosis date: the original M3 paper also validated the index performance using a one-year lookback.) The file is specified as the argument in\_df

The M3 index can also use cancer registry data (e.g. the NZ Cancer Registry) to supplement information on cancers that are not recorded on hospital admissions (or not recorded in sufficient detail). This includes classifying primary tumours by site, and also coding for metastatic cancers. *Currently this is not implemented in the R function.*

*Usage as per example code*

For example, the main input datafile may be in the dataframe called my\_diagnosis\_df.

## Input format:

At present, the code required the input data to be in **long** format. The input file can contain columns in addition to the identifying variable[s] and diagnostic code column[s], but these additional columns are stripped out during the processing step (see Output Format below). The user will need to merge the output file with scored conditions (from this function) to the per-patient information (e.g. age, gender, cancer stage) if this is needed.

*Identifying variables*

The names of the identifying variables should be specified in the id\_cols = argument of the function call. In the example above, "patient\_ID" is passed to the function as a single ID variable. If you have two or more variables (that are required for unique identification) you can pass this to the argument as a vector of strings (e.g. c("hospital\_ID", "patient\_ID") if patient numbers are nested within hospital identifiers.)

*Please note that the multiple identifier columns functionality has not been thoroughly tested: it should work as intended, but a fall-back position is to create a single identifying variable in your dataset (by concatenating two string(s) or combining a string and numeric variable) and use this single variable for the* id\_cols *argument.*

*Long format:*

In long format data, a single column contains all diagnoses, usually with separate identifier[s] for each patient and for the hospital admission for which each diagnoses was recorded (see *mockdata\_example.xlsx,* in the Long layout sheet) The name of the column holding this variable needs to be specified in clinical\_code\_col= (default is CLIN\_CD).

## Output data filename and format:

***Please note the function currently only returns rows for people with at least one M3 condition in the “diagnostic codes” input. Individuals with no identified conditions will not be returned in the output, and will need to be assigned individual condition indicators of zero and an overall score of zero. This may change in a future update.***

Don’t forget to store the results of running the function as an object.

The format of the output dataframe is one row per person identified from the Input File (unique people as defined by identifying variable(s) in that file: **see bolded note above**) with the identifying variable[s] as first columns, followed by:

**Identifying variable[s]**: the identifying variable[s] as specified in the original call to the function.

**(optional output, 61 variables prefixed with “M3\_”):** sixty-one columns for presence/absence of each health condition as classified by the M3 index (including those five conditions subsequently weighted as zero). These are coded as 0 for “condition not found” and 1 for “condition found.” All of these variables are prefixed with “M3\_”

**M3Score:** the M3 Index score (continuous scale.) We generally recommend using this variable for adjustment in analytical models (such as logistic regression, Cox proportional hazards models), and have typically used it ourselves using restricted cubic splines (RCS) when adjusting for comorbidity as a confounder. Within R, these spline-type functions can be created and deployed using packages like splines:: or rms::

## Using the Output data:

Once the Output dataset is created, it can then be merged back onto a “core analysis dataset” that would typically have one row per patient. The following description gives a typical example for a post-surgery survival analysis:

The dataset has columns for the identifying variable[s] (e.g. patient ID number), age at diagnosis, indication/reason for surgery, date of death and cause of death (if patient died) and time last known to be alive (if patient still alive at last follow-up time.) This file would also contain additional variables (e.g. in a randomised controlled trial, a variable indicating intervention/control arm; in an observational study, a main exposure of interest or additional confounding variables.)

The output datafile for the M3 index can then be merged back onto the surgery survival dataset (matching on the identifying variable[s]) which will add at least one columns (M3 index score – columns for the individual M3 conditions with 1/0 coding will be added if these were kept during the coding steps, which is the default.)

# Calculating the C3 index

The paper describing the creation and validation of the C3 index (Sarfati et al., 2014b) is:

**Sarfati, D., Gurney, J., Stanley, J., Salmond, C., Crampton, P., Dennett, E., Koea, J. & Pearce, N. (2014b). Cancer-specific administrative data-based comorbidity indices provided valid alternative to Charlson and National Cancer Institute Indices. *Journal of Clinical Epidemiology*, 67(5), 586-595.**

**doi: 10.1016/j.jclinepi.2013.11.012**

The primary function is c3index\_scoring() – see the function file for a list of all options. The main options are demarcated below (with an example) for a typical usage scenario for a cohort of breast cancer patients.

If you are dealing with a cohort of patients with different cancer sites, then you should process each cancer site separately with this top-level function.

(While there is some grouping of the C3 index weightings later on – e.g. common weights for gynaecological cancers –the initial processing needs to be site specific, e.g. uterine cancer and ovarian cancer group data need to be passed through the function in two separate runs.)

The paper by Sarfati et al. 2014b also describes site-specific weightings, where the indices were based on cancer site-specific estimates of non-cancer mortality. In our subsequent work we determined that the all-sites version of the index performs at roughly the same level, and so this is the index included in this code release. If you are interested in the site-specific weightings you can contact [james.stanley@otago.ac.nz](mailto:james.stanley@otago.ac.nz) for code to produce the tumour site-specific C3 indices.

Example call to function for breast cancer dataset

test\_C3 <- c3index\_scoring(in\_df=all\_adm, in\_df\_precancer=prior\_adm,

cancersite="COLON",

clinical\_code\_col = "CLIN\_CD",

id\_cols = "patient\_ID")

## Input Files:

*Overview*

The input file should contain (at a minimum) identifying variable(s) that indicate individual people, and one or more columns with ICD-10 codes for diagnoses recorded for hospital admissions.

This file should be pre-prepared by the user so that it contains only those diagnoses that fall within the lookback eligibility period (for C3, we consider this as the five year period up to and including the index hospitalisation/diagnosis date.) The file is specified as the function argument in\_df=

The C3 index codes several conditions (myocardial infarction, congestive heart failure, pulmonary embolism, anxiety and behavioural disorders, anaemia, hypertension, and cardiac arrhythmias) as present *only* if they were recorded *prior* to cancer diagnosis/index admission date. This is to exclude the possibility that such conditions were “complications of the primary disease or its treatment.” (see p. 558 of Sarfati et al. 2014b for more detail.)

This requires the end-user to prepare a second data file from the initial diagnostic code input file so that it only includes diagnostic records that come before the cancer diagnosis/index admission date. This secondary processing step is then used to code these specific conditions (as listed above.)

*Usage as per example code*

For example, the main input dataframe may be called **all\_adm**; the secondary datafile (restricted by the user to cover diagnostic codes for admissions prior to cancer diagnosis) is created by the user and stored for instance as **prior\_adm**.

## Input format:

At preent, the input data needs to be in **long** format. The input file can contain columns in addition to the identifying variable[s] and diagnostic code column[s], but these additional columns are stripped out during the processing step (see Output Format below). The user will need to merge the output file with scored conditions (from this function) to the per-patient information (e.g. age, gender, cancer stage) if this is needed.

*Identifying variables*

The names of the identifying variables should be specified in the id\_cols= argument of the function call as a string. In the example above, "patient\_ID" is passed to the function as a single ID variable. If you have two or more variables (that are required for unique identification) you can pass this to the argument as a vector of strings (e.g. c("hospital\_ID", "patient\_ID"), if patient numbers are nested within hospital identifiers.)

*Please note that the multiple identifier columns functionality has not been thoroughly tested: it should work as intended, but a fall-back position is to create a single identifying variable in your dataset (by concatenating two string(s) or combining a string and numeric variable) and use this single variable for the* id\_cols *argument.*

*Long format:*

In long format data, a single column contains all diagnoses, usually with separate identifier[s] for each patient and for the hospital admission for which each diagnoses was recorded (see *mockdata\_example.xlsx,* in the Long layout sheet) The name of the column holding this variable needs to be specified in clinical\_code\_col= (default is "CLIN\_CD")

## Output data filename and format:

***Please note the function currently only returns rows for people with at least one M3 condition in the “diagnostic codes” input. Individuals with no identified conditions will not be returned in the output, and will need to be assigned individual condition indicators of zero and an overall score of zero. This may change in a future update.***

Don’t forget to store the results of running the function as an object for subsequent use!

The format of the output file is one row per person identified from the Input File (unique people as defined by identifying variable(s) in that file: **see bolded note above**) with the identifying variable[s] as first columns, followed by:

**Identifying variable[s]**: the identifying variable[s] as specified in the original call to the function.

**(optional output, 42 variables prefixed with “C3\_”):** forty-two columns for presence/absence of each health condition as classified by the C3 index. These are coded as 0 for “condition not found” and 1 for “condition found.” All of these variables are prefixed with “C3\_”

**C3score\_allsites:** the C3 Index score for all sites (continuous scale.) We generally recommend using this variable for adjustment in analytical models (such as logistic regression, Cox proportional hazards models), and have typically used it ourselves using restricted cubic splines (RCS) when adjusting for comorbidity as a confounder. Within R, these spline-type functions can be created and deployed using packages like splines:: or rms::

**C3cat\_allsites:** the C3 Index score for all sites, *categorised as per Sarfati et al. 2014b.* Note that we only recommend using this categorised variable for descriptive statistics describing comorbidity profiles per group, and not as an exposure or covariate in analytical models. The categorisation scheme we have used is ‘0’ (C3 Index score < =0), ‘1’ (0 < score < =1), ‘2’ (1 < score < =2), ‘3’ (score > 2) (see e.g. Sarfati et al., 2014a) – the original paper split this last category into ‘3’ (2 < score <= 3) and ‘4’ (score > 3) for descriptive reporting (Table 3, Sarfati et al., 2014b.) This code uses the 0-3 categorisation.

## Using the Output data:

Once the Output dataset is created, it can then be merged back onto a “core analysis dataset” that would typically have one row per patient. The following description gives a typical example for a cancer survival analysis:

The dataset has columns for the identifying variable[s] (e.g. patient ID number), age at diagnosis, site of cancer (e.g. breast, colon), cancer stage at diagnosis, date of death and cause of death (if patient died) and time last known to be alive (if patient still alive at last follow-up time.) This file would also contain additional variables (e.g. in a randomised controlled trial, a variable indicating intervention/control arm; in an observational study, a main exposure of interest or additional confounding variables.)

The output datafile for the C3 index can then be merged back onto the cancer survival dataset (matching on the identifying variable[s]) which will add at least two columns (C3 index score, and C3 index descriptive category – individual conditions with 1/0 coding will be added if these were kept during the coding steps.)

## References (in alphabetical order)

Charlson, M. E., Pompei, P., Ales, K. L., & MacKenzie, C. R. (1987). A new method of classifying prognostic comorbidity in longitudinal studies: development and validation. J Chronic Dis, 40(5), 373-383.

Desquilbet, L., & Mariotti, F. (2010). Dose-response analyses using restricted cubic spline functions in public health research. Statistics in Medicine, 29(9), 1037-1057. doi: 10.1002/sim.3841

Elixhauser A, Steiner C, Harris DR, Coffey RM. Comorbidity measures for use with administrative data. Med Care. 1998;36(1):8-27.

Quan, H., Sundararajan, V., Halfon, P., Fong, A., Burnand, B., Luthi, J. C., Saunders, L. D., Beck, C. A., Feasby, T. E., Ghali, W. A. (2005). Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data. Med Care, 43(11), 1130-1139.

Sarfati, D., Gurney, J., Stanley, J., & Koea, J. (2014a). A retrospective cohort study of patients with stomach and liver cancers: the impact of comorbidity and ethnicity on cancer care and outcomes. *BMC Cancer*, 14, 821. doi: 10.1186/1471-2407-14-821

Sarfati, D., Gurney, J., Stanley, J., Salmond, C., Crampton, P., Dennett, E., Koea, J. & Pearce, N. (2014b). Cancer-specific administrative data-based comorbidity indices provided valid alternative to Charlson and National Cancer Institute Indices. *Journal of Clinical Epidemiology*, 67(5), 586-595.

doi: 10.1016/j.jclinepi.2013.11.012

Stanley J, Sarfati D. (2017) The new measuring multimorbidity index predicted mortality better than Charlson and Elixhauser indices among the general population. Journal of Clinical Epidemiology 2017;92:99-110. DOI: 10.1016/j.jclinepi.2017.08.005

van Walraven C, Austin PC, Jennings A, Quan H, Forster AJ. A modification of the Elixhauser comorbidity measures into a point system for hospital death using administrative data. Med Care. 2009;47(6):626-33.

## Technical notes

The R functions herein were based on earlier SAS macros developed in SAS v9 (versions 9.1 to 9.4), but have been re-written as much as possible to maximise efficiency (within the limits of my programming skills!)

## Acknowledgements

This ReadMe file was written by James Stanley.

The C3 and Multimorbidity projects were funded by NZ Health Research Council (HRC) research project grants with Diana Sarfati as Principal Investigator. Clare Salmond and Jason Gurney wrote much of the original C3 project SAS code for classifying conditions and applying weights; James Stanley and Jane Zhang re-wrote parts of this SAS code for the Multimorbidity project, which is used for both indices in this release, and the R code was written and is maintained by James Stanley.