# Package ‘dalycare’

October 17, 2023

**Title** Danish Lymphoid Cancer Research

**Priority** NA

**Version** 1.1.1

**Date** 2025-03-10

**Depends** R (>= 4.2.0), tidyverse, lubridate, haven, readxl, data.table, Survival, survminer, Publish, ggpubr, RSQLite, DBI, odbc, RPostgres, RPostgreSQL, docstring, insight.

**Imports** Codes\_NPU

**LazyData** ?

**LazyDataCompression** ?

**ByteCompile** ?

**Description** Contains definitions and grouping of Danish electronic health data from SDS, RKKP, and SP.

**License** ?

**URL** NA

**NeedsCompilation** NA

**Author** Christian Brieghel [aut, cre], Casper Møller Frederiksen [ctb, trl], Mikkel Werling [cre , ctb, trl]

**Maintainer** Christian Brieghel

**Repository**: <https://github.com/RH-CLL-LAB/dalycare_package>

**Date/Publication**: 2025-02-24

## Index

Package 'dalycare' is loaded from NGC in RStudio as:

source('/ngc/projects2/dalyca\_r/clean\_r/load\_dalycare\_package.R')

## Cleaners

##### clear\_ram

Description

Clears Global environment and frees RAM from NGC/dalycare

Usage

clear\_ram()

##### clean\_RKKP\_LYFO

Description

Cleans the dataset RKKP\_LYFO. Works only for LYFO version 20 or higher, please see [rkkp-documentation](https://www.rkkp-dokumentation.dk/Public/Default.aspx?msg=ChooseDB&error=WrongParm)

Usage

RKKP\_LYFO\_CLEAN = RKKP\_LYFO %>% clean\_RKKP\_LYFO()

##### clean\_RKKP\_LYFO\_SNOMED

Description

Cleans SNOMED codes in RKKP\_LYFO. Works only for LYFO version 20 or higher, please see [rkkp-documentation](https://www.rkkp-dokumentation.dk/Public/Default.aspx?msg=ChooseDB&error=WrongParm)

Usage

RKKP\_LYFO %>%

mutate(icd10 = clean\_RKKP\_LYFO\_SNOMED(snomed = Reg\_WHOHisto))

##### clean\_RKKP\_CLL

Description

Cleans the dataset RKKP\_CLL. Works only for CLL registry version 15 or higher, please see [rkkp-documentation](https://www.rkkp-dokumentation.dk/Public/Default.aspx?msg=ChooseDB&error=WrongParm)

Usage

RKKP\_CLL\_CLEAN = RKKP\_CLL %>% clean\_RKKP\_CLL()

##### clean\_RKKP\_DAMYDA

Description

Cleans (or translates) the dataset RKKP\_DAMYDA. Works only for DAMYDA version 18 or higher, please see [rkkp-documentation](https://www.rkkp-dokumentation.dk/Public/Default.aspx?msg=ChooseDB&error=WrongParm)

Usage

RKKP\_DAMYDA\_CLEAN = RKKP\_DAMYDA %>% clean\_ RKKP\_DAMYDA()

##### clean\_RKKP\_DAMYDA\_SNOMED

Description

Cleans SNOMED (or translates) codes in RKKP\_DAMYDA. Works only for DAMYDA version 20 or higher, please see [rkkp-documentation](https://www.rkkp-dokumentation.dk/Public/Default.aspx?msg=ChooseDB&error=WrongParm)

Usage

RKKP\_DAMYDA %>%

mutate(icd10 = clean\_RKKP\_DAMYDA\_SNOMED(snomed = Reg\_WHOHisto)

##### clean\_abbreviations

Description

Replaces commonly used Danish abbreviations containing punctuation to allow for better separation of free text into complete sentences.

E.g. 'f.eks. ' to 'f\_eks\_' pattern.

Caveat: time lapse with large datasets: subset data before use.

Usage

SP\_Journalnotater\_Del1 %>%

mutate(notat\_text = clean\_abbreviations(notat\_text))

##### clean\_lab\_values

Description

Cleans and converts common laboratory values with correct units based on NPU codes.

E.g. B2M nmol/l converts to mg/l.

Usage

load\_npu\_common()

LAB\_clean = load\_biochemistry(NPU.GROUP.INFECTION) %>%

clean\_lab\_values()

##### clean\_SDS\_t\_mikro

Description

Cleans and aggregates pathology free-text descriptions from the t\_mikro dataset.

E.g. lines 1 “Biopsy examined”, 2 “by immunohistochemistry”, and 3 “shows follicular lymphoma” convert to “Biopsy examined by immunohistochemistry shows follicular lymphoma”.

Usage

t\_mikro\_clean = SDS\_t\_mikro %>%

clean\_SDS\_t\_mikro()

##### clean\_SKS\_codes\_B

Description

Left joins SKS B code text to SKS B codes.

Usage

t\_sksube\_clean = SDS\_t\_sksube\_subset %>%

clean\_SKS\_codes\_B()

## Loaders

##### load\_dataset

Description

Loads data directly from the DALY-CARE database when specifying dataset(s).

*Dataset* may be specified as a vector of datasets.

Returns a complete list of dataset options when *dataset* is NULL (default).

Imports subset of dataset(s) when specifying *sample* as a vector of *patientid(s).*

Also imports subset of dataset on other existing variables specifying *filter* argument.

Usage

load\_dataset() #Returns a list of available *datasets*

load\_dataset(c('patient', 'RKKP\_CLL\_CLEAN')) # loads both

load\_dataset('RKKP\_DAMYDA', value = sample(PATIENT$patientid, 100)) #only sample

load\_dataset('SP\_OrdineretMedicin', value = c('J06BA02', 'J01CE01'), column = 'atc')

##### load\_datasets\_head

Description

Loads the head of all PERSIUNE, RKKP, SDS and SP datasets to get an overview of data structure.

Specify nrows in the head argument (e.g. head = 20).

Usage

load\_datasets\_head() #Returns the header of all *datasets* to your global environment

##### load\_all\_variables

Description

Loads all variables of all DALY-CARE datasets: Please see Table S2 and Appendix3

Usage

load\_all\_variables() %>% print\_data()

##### load\_all\_data

Description

Loads all key variables from key datasets on a subset of patients

Usage

ALL\_DATA = load\_all\_data()

##### load\_dalycare\_icd10

Description

Loads definitions of DALY-CARE entities based on ICD10 diagnoses into vectors located in your Global Environment in R: Please see Table S7

Usage

load\_dalycare\_icd()

FL = t\_dalycare\_diagnoses %>%

filter(diagnosis %in% ICD10.FL)

##### load\_blood\_culture\_SP

Description

Loads blood cultures from SP\_AlleProvesvar

Usage

BC = load\_blood\_culture\_SP()

##### load\_SKS\_antineoplastic

Description

Loads a list of vectors containing antineoplastic SKS codes used to treat LC to the Global Environment. You may specify individual codes such as SKS.ibrutinib (ie. ibrutinib), which are grouped SKS.CLL\_targeted (i.e. ibrutinib, zanubrutinib, venetoclax and idelalisisb; note missing code for acalabrutinib in SKS), and further grouped into SKS.CLL\_treatment (i.e. chemo-, immuno-, and targeted-therapy). Similar logic exists for lymphomas and multiple myeloma. SKS treatment codes are found in the table CODES\_B under *core – lookuptables* and downloaded from <https://medinfo.dk/sks/dump.php>

Usage

load\_SKS\_antineoplastic()

SKS.ibrutinib # returns BWHA427

SKS.CLL\_targeted #returns BWHP114, BWHA427, BWHA428, BWHA438

# Use SKS codes to load\_dataset() subset

load\_dataset('SDS\_t\_sksube', c(SKS.ibrutinib, SKS.venetoclax), 'c\_opr')

SDS\_t\_sksube\_subset$c\_opr %>% table()

load\_dataset('SDS\_procedurer\_andre', SKS.MM\_proteasome, 'procedurekode')

SDS\_procedurer\_andre$procedurekode %>% table()

##### load\_npu\_common

Description

Loads a list of vectors containing common NPU codes to Global Environment.

You may specify individual codes such as NPU.LYM (ie. lymphocytes) or groups of NPU codes such as GROUP.NPU.CBC (i.e. complete blood count) or NPU.GROUP.MYELOMA (i.e. standard myeloma blood test set).

Usage

load\_npu\_common()

NPU.HGB # returns NPU02319

# Use NPUs to load\_dataset() subset

load\_dataset('SDS\_lab\_forsker', c(NPU.B2M, NPU.LYM), 'analysiscode')

SDS\_lab\_forsker\_subset$analysiscode %>% table()

##### load\_biochemistry

Description

Loads dataset containing biochemistry from SDS\_lab\_forsker.

'labs' must contain NPU codes, e.g. from lists from load\_npu\_common()

Usage

LAB\_df = load\_biochemistry(c(NPU.B2M, NPU.LDH))

BSI\_df = load\_biochemistry(NPU.BSI) #Blood cultures

#assign data as SDS\_lab\_forsker\_subset into Global Environment

load\_biochemistry(labs = NPU.GROUP.MSPIKE, assign = TRUE)

##### go\_live

Description

Loads SP (EPIC) go live dates for the three hospitals HGH, Herlev; Rigshospitalet; and SUH, Roskilde.

Usage

go\_live()

CLL\_clean = RKKP\_CLL %>% clean\_RKKP\_CLL() %>% left\_join(go\_live)

CLL\_clean$date\_golive

## Definers

##### filter\_first\_diagnosis

Description

Defines first DALY-CARE diagnosis from 't\_dalycare\_diagnoses' as the earliest occurrence and calculates KM years from table 'patient'. Note that filter\_first\_diagnosis comes with a caveat, because it uses all diagnoses regardless of origin to define the first occurring diagnosis. Thus, patients may in fact have been diagnosed years prior to 2002, and if they are then admitted with a LC diagnosis after 2002, this date admission diagnosis will define their first diagnosis.

Usage

load\_ dataset('t\_dalycare\_diagnoses', 'patient') #loads all DALY-CARE diagnoses

PCD = t\_dalycare\_diagnoses %>%

filter(tablename %in% c('t\_pato', 'DaMyDa', 't\_tumor')) %>%

filter\_first\_diagnosis('DC90') #includes any DC90.x

CLL = t\_dalycare\_diagnoses %>%

filter(tablename %in% c('t\_pato', 'RKKP\_CLL', 't\_tumor')) %>% # only diagnoses from

filter\_first\_diagnosis('DC911', str\_contains = FALSE)

MZL = t\_dalycare\_diagnoses %>%

filter(tablename %in% c('t\_pato', 'RKKP\_LYFO', 't\_tumor')) %>% #

filter\_first\_diagnosis(c('DC830C', 'DC830D', 'DC884', 'DC884A', 'DC884B', 'DC884C'))

load\_dalycare\_icd10() # loads a list of vectors with ICD10 LC diagnoses

MZL = t\_dalycare\_diagnoses %>%

filter(tablename %in% c('t\_pato', 'RKKP\_LYFO', 't\_tumor')) %>% #

filter\_first\_diagnosis(ICD10.MZL, str\_contains = FALSE) #all MZL diagnoses

SLL = t\_dalycare\_diagnoses %>%

filter(tablename %in% c('t\_pato', 'RKKP\_LYFO', 't\_tumor')) %>% #

filter\_first\_diagnosis('DC830', str\_contains = FALSE) #matches 'DC830'

RICHTER = t\_dalycare\_diagnoses %>%

filter(tablename %in% c('t\_pato', 'RKKP\_LYFO', 'RKKP\_CLL', 't\_tumor')) %>% #

filter\_first\_diagnosis(c('DC833', 'DC911'), multiple = 'both') #matches both

##### right\_truncation (aka truncate\_time\_to\_event)

Description

Right truncates date of last follow-up as date\_event\_death\_fu and calculates time\_to\_event and event\_competing as 0 (cens), 1 (event), and 2 (comepeting). This should always be checked in time-to-event analyses to avoid immortality bias, especially when linking data.

Usage

load\_dataset('t\_dalycare\_diagnoses', 'patient') #loads all DALY-CARE diagnoses

CLL = t\_dalycare\_diagnoses %>%

filter\_first\_diagnosis('DC911', string\_contains = FALSE) %>%

left\_join(your\_event\_data %>% select(patientid, date\_event), by = 'patientid') %>% #expects your\_event\_data in wide format

right\_truncation(date\_event, #date of event. Expects NA for non-events

date\_start = date\_diagnosis, # prediction date, e.g. date\_diagnosis

date\_truncation = '2023-1-1') #last event as character

library(cmprsk) #for competing risk analyses

fit = cuminc(ftime = CLL$time\_to\_event,

fstatus = CLL$event\_competing,

group = 1) #group stratifies

timepoints(fit, c(1,5,10))

library(Publish) #for plotting competing risk analyses with no. at risk.

aj = prodlim(Hist(time\_to\_event, event\_competing)~1, data=CLL)

plot(aj)

#”Error in plot.new()” may be rectified by: par(mar=c(1,1,1,1))

##### scr\_low\_48h

Description

Defines lowest serum creatinine (scr) within 48 hours using lab\_forsker data.

SDS\_lab\_forsker data should be filtered to contain creatinine only (NPU.KREA) to avoid time-lapse. Used to define acute kidney injury (AKI).

Usage

load\_npu\_common()

load\_dataset('SDS\_lab\_forsker', c(NPU.KREA), 'analysiscode') #loads creatinine

DATA\_scr\_low\_48h = SDS\_labforsker\_subset %>%

mutate(

cpr\_enc = patientid,

date\_time = as.numeric(seconds(as.POSIXct(paste(samplingdate, samplingtime)))),

i.scr\_inhos = 0

) %>%

scr\_low\_48h()

##### scr\_low\_7d

Description

Defines lowest serum creatinine (scr) within 7 days using lab\_forsker data.

SDS\_lab\_forsker data should be filtered to contain creatinine only (NPU.KREA) to avoid time-lapse.

Usage

load\_npu\_common()

load\_dataset('SDS\_lab\_forsker', c(NPU.KREA), 'analysiscode') #loads creatinine

DATA\_scr\_low\_48h = SDS\_labforsker\_subset %>%

mutate(

cpr\_enc = patientid,

date\_time = as.numeric(seconds(as.POSIXct(paste(samplingdate, samplingtime)))),

i.scr\_inhos = 0

) %>%

scr\_low\_7d()

##### scr\_base\_median

Description

Defines baseline serum creatinine (BL scr) a rolling median using lab\_forsker data.

SDS\_lab\_forsker data should be filtered to contain creatinine only (NPU.KREA) to avoid time-lapse.

Usage

load\_npu\_common()

load\_dataset('SDS\_lab\_forsker', c(NPU.KREA), 'analysiscode') #loads creatinine

DATA\_scr\_low\_48h = SDS\_labforsker\_subset %>%

mutate(

cpr\_enc = patientid,

date\_time = as.numeric(seconds(as.POSIXct(paste(samplingdate, samplingtime)))),

i.scr\_inhos = 0

) %>%

scr\_base\_median()

##### AE\_AKI

Description

Defines acute kidney injury based on a 1.5x increase from the baseline serum creatinine (scr\_base\_median) within 7 days (scr\_low\_7d) or an absolute scr increase of 26.5 µmol/L within 48 hours (scr\_low\_48h) using lab\_forsker data.

SDS\_lab\_forsker data should be filtered to contain creatinine only (NPU.KREA) to avoid time-lapse.

Usage

load\_dataset('SDS\_lab\_forsker', c(NPU.KREA), 'analysiscode') #loads creatinine

CREATININE\_clean = SDS\_labforsker\_subset %>% clean\_lab\_values()

AKI = CREATININE\_clean %>% AE\_AKI(value = value2)

Citation

Carrero JJ et al. Kidney Int. 2023 Jan;103(1):53-69.

##### AE\_infection

Description

Defines infections based on duration of iv. antimicrobial therapy (AB\_min\_duration [days]: default 1.00 days) and days between AB separating 2 infectious events (days\_between\_separating\_infections: default 7.00 days) from SP antimicrobial data using SP\_AdministreretMedicin data after filtering AB only; first using ATC\_AB().

Usage

load\_dataset ('patient')

load\_dataset('SP\_AdministreretMedicin', sample(patient$patientid, 1000))

SP\_AB = SP\_AdministreretMedicin\_subset %>% ATC\_AB()

SP\_infections = SP\_AB %>% AE\_infection()

Citation

Brieghel C et al. Polypharmacy. 2025 (*work in progress*).

##### CTCAE\_lab

Description

Defines CTC adverse events (AE) from biochemistry. Works only with lab\_forsker data.

SDS\_lab\_forsker data should be filtered to contain NPU of interest to avoid time-lapse.

E.g. May calculate 'ANEMIA', 'THROMBOCYTOPENIA', 'DIC', and 'HEMOLYSIS'.

Usage

load\_npu\_common()

HGB = load\_biochemistry(NPU.HGB) %>% clean\_lab\_values()

ANEMIA\_AE = HGB %>%

CTCAE\_lab() %>%

select(patientid, ANEMIA.GRADE, everything())

HEMOLYSIS = load\_biochemistry(NPU.GROUP.HEMOLYSIS) %>%

clean\_lab\_values()

# expect time-lapse for large samples, consider down sampling

HEMOLYSIS\_AE = HEMOLYSIS %>%

CTCAE\_lab() %>%

select(patientid, HEM.GRADE, everything())

##### TX\_group

Description

Groups treatment protocols into meaningful groups as class characters.

Usage

SP\_Behandlingsplaner\_del1 %>% TX\_group() %>% pull(TX\_group)

##### filter\_virus

Description

Subsets RSV, SARS-CoV-2 (SARS) and seasonal influenza (FLU) into class character (type) and result.

Usage

SP\_Bloddyrkning\_del1 %>% filter\_virus() %>% select(patientid, type, result)

Citation

Niemann et al. *Blood*. Aug 4 2022;140(5):445-450.

##### filter\_sentence

Description

Subsets all free-text sentences (i.e. from \\. to \\.) containing pattern.

Caveat: Free text often contains punctuation such as abbreviation causing separation; please see clean\_abbreviations()

Usage

SP\_Journalnotater\_del1 %>% filter\_sentence(notat\_text, 'SAGM')

SDS\_t\_mikro\_ny %>% filter\_sentence(v\_fritekst, 'EBER')

##### ATC\_polypharmacy

Description

Calculates number of 1st to 5th level ATC codes per patient and defines polypharmacy as ≥5 drug classes.

Usage

SDS\_epikur %>% ATC\_polypharmacy(level = 3) %>% pull(Polypharmacy)

Citation

Brieghel et al. ASH annual meeting 2023. P5133

##### COD2

Description

Groups cause of death (COD) ICD10 codes into meaningful groups. Prioritizes infections.

Usage

SDS\_t\_dodsaarsag\_2 %>%  COD2()

Citation

Rotbain et al. *Leukemia*. 2021;35(9):2570-2580.

##### CCI

Description

Calculates Charlson comorbidity index (CCI) scores from ICD10 codes.

exclude\_CLL\_score = FALSE (default) includes the DC911 score, if present.

include\_LC\_score = FALSE (default) calculates the LC score only if present.

Usage

SDS\_t\_adm %>% CCI(icd10 = c\_adiag) %>% select(patientid, CCI.score, CCI.2011.update)

view\_diagnoses\_all %>% CCI() %>% select(patientid, CCI.score, CCI.2011.update)

Citation

Quan et al. Med Care. 2005;45:1130-9 as CCI.score

Quan et al. Am J Epidemiol. 2011;173:676-82 for CCI.2011.update

##### CLL\_CI

Description

Calculates CLL comorbidity index (CLL-CI) scores from vascular, GI, and endocrinology defined SKS codes (LPR), ATC codes (EPIKUR), and ICD10 codes (diagnoses\_all).

Usage

CLL\_cohort = t\_dalycare\_diagnoses %>%

filter\_first\_diagnosis('DC911', str\_contains = FALSE) #create CLL cohort

CLL\_CI\_cohort = CLL\_cohort %>%

CCI\_CI() # Input only requires variable with 'patientid'

Citation

Rotbain et al. *Blood Adv*. 2022;6(8):2701-6

##### NMI

Description

Calculates Nordic Multimorbidity Index (NMI) scores from ICD10 and ATC codes before date of diagnosis (as date\_diagnosis).

Usage

your\_cohort = t\_dalycare\_diagnoses %>%

filter\_first\_diagnosis(ICD10.CLL) %>%

NMI() %>%

select(patientid, NMI\_score)

Citation

Kristensen et al. *CLEP*. 2022;14:567-79

##### ATC\_AB

Description

Subsets and groups all antimicrobials.

Usage

SDS\_epikur %>% ATC\_AB()

SP\_AdministreretMedicin %>% ATC\_AB()

##### ATC\_hypertensives

Description

Subsets and groups all antihypertensive drugs.

Usage

SDS\_epikur %>% ATC\_hypertensives()

SP\_Administreret\_Medicin %>% ATC\_hypertensives ()

##### ATC\_opioids

Description

Subsets and groups all opioids.

Usage

SDS\_epikur %>%  ATC\_opioids()

SP\_Administreret\_Medicin %>% ATC\_opioids()

##### qSOFA

Description

Calculates qSOFA scores from vital values assuming that AVPU less than alert (A) replaces GCS < 15.

Usage

SP\_VitaleVaerdier %>% qSOFA() %>% pull(qSOFA)

##### BMI

Description

Calculates body mass index (BMI) and body surface area (BSA) from vital values.

Usage

SP\_VitaleVaerdier %>% BMI() %>%

select(patientid, BMI, BSA\_DuBois, BSA\_Mosteller)

##### BSA

Description

Calculates body mass index (BMI) and body surface area (BSA) from vital values.

Usage

SP\_VitaleVaerdier %>% BSA() %>%

select(patientid, BMI, BSA\_DuBois, BSA\_Mosteller)

##### transform\_2\_ERIC

Description

Transforms DALY-CARE data to the standard format for submission of data to projects within European Research Initiative on CLL (ERIC) and the ERIC database. Data defined from RKKP\_CLL, RKKP\_LYFO (for SLL cases), LAB\_IGHVIMGT, patient, t\_dalycare\_diagnoses, diagnoses\_all (comorbidity and second malignancy), SDS\_t\_doedsaarsag and CLL\_TREAT\_CLEAN (only if 2nd line treatment is missing from RKKP).

write\_xlsx = TRUE. Writes an Excel file with 2 sheets and appreciated by ERIC

pseudononymize = TRUE. Creates pseudononymized `Patient Lab id`.

NB! Always deselect the DALY-CARE patientid before sharing data with Thomas Chatzikonstantinou via secure warehouse: [thomas.chatzikonstantinou@certh.gr](mailto:thomas.chatzikonstantinou@certh.gr)

Usage

CLL\_SLL\_cohort = t\_dalycare\_diagnoses %>%

filter\_first\_diagnosis(c('DC911', 'DC833'), str\_contains = FALSE) #create CLL/SLL #cohort

ERIC\_data = CLL\_SLL\_cohort %>%

transform\_2\_ERIC() # Input only requires variable with 'patientid'

ERIC\_data = CLL\_SLL\_cohort %>%

transform\_2\_ERIC(write\_xlsx= TRUE, pseudononymize = TRUE) # writes Excel file with #pseudo IDs

Citation

Chatzidimitrou et al. *Hemasphere*. 2020;4(5):e425

##### write\_utable

Description

Writes utables as publishable csv-files to your work directory

table\_n = 1, Writes table “Table\_1” etc.

Usage

getwd()

CLL\_clean = RKKP\_CLL %>% clean\_RKKP\_CLL()

table1 = utable(sex ~ Q(age) + binet, CLL\_clean)

write\_utable(table1)

##### CLL\_IPI

Description

Calculates CLL-IPI risk as class factor.

Usage

RKKP\_CLL\_CLEAN %>% CLL\_IPI() %>% pull(CLL.IPI) %>% table()

Citation

da Cunha-Bang et al. *Blood*. 2016;128(17):2181-3.

##### CLL\_WONT

Description

Calculates CLL-WONT risk as class factor. Needs ALC (NPU02636) and LDH (NPU19658; NPU19978; NPU19975) from e.g. SDS\_lab\_forsker. Consider skipping data preparation.

Usage

# Data preparation

load\_npu\_common()

LAB = load\_biochemistry (labs = c(NPU.LYM, NPU.LDH)) %>%

clean\_lab\_values()

ALC = LAB %>%

filter(NPU %in% NPU.LYM) %>%

transmute(patientid, date\_ALC = samplingdate, ALC = value2)

LDH = LAB %>%

filter(NPU %in% NPU.LDH) %>%

transmute(patientid, date\_LDH = samplingdate, LDH = value2)

# Data preparation continued…

RKKP\_CLL\_WITH\_ALC\_AND\_LDH = RKKP\_CLL\_CLEAN %>%

left\_join(ALC, by = 'patientid') %>%

left\_join(LDH, by = 'patientid') %>%

mutate(time\_ALC = diff\_days(Date\_diagnosis, date\_ALC),

time\_LDH = diff\_days(Date\_diagnosis, date\_LDH)) %>%

filter(time\_ALC <= 0, time\_ALC >= -90,

time\_LDH <= 0, time\_LDH >= -90) %>%

group\_by(patientid) %>%

arrange(patientid, desc(time\_ALC), desc(time\_LDH)) %>%

slice(1) %>%

ungroup()

# CLLWONT calculation

RKKP\_CLL\_WITH\_ALC\_AND\_LDH %>%  CLL\_WONT() %>%

pull(CLLWONT) %>% table()

Citation

Brieghel et al. *Eur J Haematol*. May 2022;108(5):369-378.

Brieghel et al. *Blood Adv*. 2024;8(16):4449-56.

##### NCCN\_IPI

Description

Calculates NCCN-IPI risk for DLBCL as class factor.

NB! Input is complex and not generalizable.

Usage

RKKP\_LYFO %>% clean\_RKKP\_LYFO() %>%

NCCN\_IPI() %>% pull(NCCN\_IPI) %>% table()

Citation

Zhou et al. *Blood*. Feb 6 2014;123(6):837-42.

Jelicic et al. BJC. 2023;13(1):157.

##### MIPI

Description

Calculates MIPI risk for Mantle cell lymphoma as class factor

Usage

RKKP\_LYFO %>% clean\_RKKP\_LYFO() %>%

MIPI() %>% pull(MIPI) %>% table()

Citation

Hoster et al. *Blood*. Jan 15 2008;111(2):558-65.

##### IPS

Description

Calculates IPS risk for Hodgkin lymphoma as class factor

Usage

RKKP\_LYFO %>% clean\_RKKP\_LYFO() %>%

IPS() %>% pull(IPS) %>% table()

Citation

Hasenclever et al. *NEJM*. 1998;339:1506-14.

##### IPSSWM

Description

Calculates IPSSWM risk for Waldenström macroglobulinemia (WM) and LPL as class factor.

Usage

RKKP\_LYFO %>% clean\_RKKP\_LYFO() %>%

IPSSWM() %>% pull(IPSSWM) %>% table()

Citation

Morel et al. *Blood*. 2009;113(18):4163-70.

##### rIPSSWM

Description

Calculates rIPSSWM risk for Waldenström macroglobulinemia (WM) and LPL as class factor.

Usage

RKKP\_LYFO %>% clean\_RKKP\_LYFO() %>%

rIPSSWM() %>% pull(rIPSSWM) %>% table()

Citation

Kastritis et al. *Leukemia*. Nov 2019;33(11):2654-2661.

##### MALT\_IPI

Description

Calculates MALT-IPI risk for marginal zone lymphoma (MZL) including patients with MALT.

Usage

RKKP\_LYFO %>% clean\_RKKP\_LYFO() %>%

MALT\_IPI() %>% pull(MALT\_IPI) %>% table()

Citation

Kastritis et al. *Leukemia*. Nov 2019;33(11):2654-2661.

##### MAYO\_20\_20\_20

Description

Calculates Mayo Institute 20-20-20 risk for progression of smoldering myeloma as class factor.

Usage

RKKP\_DAMYDA\_CLEAN %>%

MAYO\_20\_20\_20() %>%

pull(MAYO\_20\_20\_20) %>%

table()

Citation

Mateos et al. *Blood cancer journal*. Oct 16 2020;10(10):102

##### R\_ISS

Description

Calculates revised ISS (R-ISS) risk for multiple myeloma as class factor.

Usage

RKKP\_DAMYDA\_CLEAN %>%  R\_ISS() %>% pull(R\_ISS) %>% table()

Citation

Palumbo et al. *J Clin Oncol*. Sep 10 2015;33(26):2863-9.

##### R2\_ISS

Description

Calculates second revised ISS (R2-ISS) risk for multiple myeloma as class factor.

Usage

RKKP\_DAMYDA\_CLEAN %>% R2\_ISS() %>% pull(R2\_ISS) %>% table()

Citation

D'Agostino et al. *J Clin Oncol*. Oct 10 2022;40(29):3406-3418.

##### RW\_ISS

Description

Calculates revised-world ISS (RW-ISS) risk for multiple myeloma as class factor.

Usage

RKKP\_DAMYDA\_CLEAN %>% RW\_ISS() %>% pull(RW\_ISS) %>% table()

## House keeping

##### is\_odd

Description

Logical output from numeric values .

Usage

sample(1:10, 5) %>% is\_odd()

##### diff\_days

Description

Calculates numeric date intervals in days.

Usage

diff\_days(date\_start, date\_end)

##### diff\_years

Description

Calculates numeric date intervals in years.

Usage

diff\_years(date\_start, date\_end)

##### filter\_str\_detect

Description

Subsets data with strings containing vector of patterns.

Usage

CLL = t\_dalycare\_diagnoses %>%

filter\_first\_diagnosis('DC911’)

load\_dataset('SP\_Behandlingsplaner\_del1', CLL$patientid, 500)

SP\_Behandlingsplaner\_del1\_subset %>%

filter\_str\_detect(protocol\_navn, c('OBI', 'VEN'))

##### str\_between

Description

Subsets string character between two patterns for text-mining purposes.

Usage

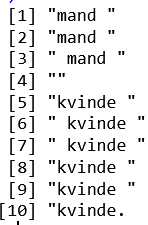
load\_dataset('SP\_Journalnotater\_del1', patient$patientid, 500)

SP\_Journalnotater\_del1\_subset %>%

filter(notat\_type=='AOP') %>%

mutate(sex = str\_between(notat\_text, 'årig', c('henvist|møder|kendt|indlægges')))%>%

pull(sex)



##### censor\_med\_keep\_first

Description

Subsets dates x days apart. Useful for censoring medication in grace period.

Usage

censor\_med\_keep\_first(date, days\_karens = 14)

Citation

Packness et al. EHA annual meeting 2022. P1596

##### cut\_year

Description

Cuts year-time into monthly intervals (e.g. 3-month intervals, by = 0.25) and outputs class factor.

Usage

Data %>% censor\_med\_keep\_first(year\_cut = cut\_year(time = Time, by = 0.25))

##### n\_patients

Description

Counts distinct patients in a dataset assuming that patients are found in 'patientid'.

Usage

patient %>% n\_patients()

##### nrow\_npatients

Description

Counts distinct patients and number of rows in a dataset assuming that patients are found in 'patientid'.

Usage

patient %>% nrow\_npatients()

##### slice\_closest\_value

Description

Slices the absolute closest value to a baseline date (date\_baseline) within time interval (interval\_days, c(-90, 0) default). Useful when adding lab values to wide format data.

Usage

load\_dataset('SP\_AlleProvesvar', NPU.HGB, 'component')

load\_dataset('patient')

patient %>%

left\_join(SP\_AlleProvesvar\_subset %>%

transmute( patientid,

date\_lab = as\_date(specimn\_taken\_time),

HGB = as.numeric(ord\_value))) %>%

slice\_closest\_value(date\_baseline = date\_diagnosis, date\_value = date\_lab)

## Plotters

##### KM\_plot

Description

Depends on library('ggplot') and library('survminer').

Plots survminer::ggsurvplot with really nice aesthetics.

Usage

CLL = t\_dalycare\_diagnoses %>%

filter\_first\_diagnosis('DC911’)

fit = survfit(Surv(time\_dx\_death, status) ~ sex, data = CLL)

KM\_plot(fit)

##### tile\_pairwise\_survdiff

Description

Depends on library('ggplot') and library('survminer').

Tiles pairwise log-rank tests from survminer::pairwise\_survdiff for visual purposes.

Usage

CLL = t\_dalycare\_diagnoses %>%

filter\_first\_diagnosis('DC911') %>%

left\_join(RKKP\_CLL\_CLEAN, by = 'patientid')

pairwise\_survdiff(Surv(time\_dx\_death, status) ~ CLL.IPI, data = CLL, p.adjust.method = 'none') %>% tile\_pairwise\_survdiff(position = 'LL', palette = c(1,2,3,4), labs = FALSE)