Package ‘dalycare’

October 17, 2023

Title Danish Lymphoid Cancer Research
Priority NA
Version 1.1.3
Date 2025-03-30
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: 2025-04-04

DALY-CARE
Danish Lymphoid Cancer Research

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

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

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

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

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

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*).

AE_hospitalization

Description

Defines hospitalization from SP_ADT_haendelser data based on mininum admission of 1 day (hospitalization_min_duration [days]: default 1.00 days).

Usage

```
load_dataset ('patient')
load_dataset('SP_ADT_haendelser', sample(patient$patientid, 1000))
SP_hospitalization = SP_ADT_haendelser_subset %>% AE_hospitalization()
```

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

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.

CNS_IPI

Description

Calculates CNS-IPI risk for DLBCL as class factor.

Usage

```
RKKP_LYFO %>% clean_RKKP_LYFO() %>%
  CNS_IPI() %>% pull(CNS_IPI) %>% table()
```

Citation

Schmitz et al. *JCO*. 2016;34(26):150-6.

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)
```

```
 [1] "mand "
 [2] "mand "
 [3] " mand "
 [4] ""
 [5] "kvinde "
 [6] " kvinde "
 [7] " kvinde "
 [8] "kvinde "
 [9] "kvinde "
[10] "kvinde.
```

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)
```

DALY-CARE
Danish Lymphoid Cancer Research

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)
```