Package 'dalycare'

October 17, 2023

Title Danish Lymphoid Cancer Research
Priority NA
Version 0.2.0
Date 2024-1-15
Depends R (>= 3.5.0), dplyr,
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 [ctb, trl]
Maintainer Christian Brieghel
Repository ?
Date/Publication ?

Index

Package "dalycare" is loaded on our NGC cloud as:
source("/ngc/projects2/dalyca_r/clean_r/load_data.R")

Cleaning
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_SP_OS

Description

Cleans the dataset SP_OS.

Usage

```
SP_OS %>% clean_SP_OS()
```

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

```
LAB_data = load_common_biochemistry(labs = "INFECTION", combine = TRUE)
LAB_clean = clean_lab_values(LAB_data)
```

clean_Date

Description

Cleans dates annotated as seconds from 1970-01-01 found in SP dates.

Usage

```
SP_VitaleVaerdier %>% clean_Date(recoded_time)
```

Load data

load_dataset

Description

Loads data directly from the DALYCARE 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_dalycare_dx

Description

Loads all diagnoses defined in DALY CARE:
C81.x-C90.x, C91.1-C91.9, C95.1, C95.7, C59.9, D47.2, D47.9B, and E85.8A.
RKKP > ICD10 > SP > SNOMED. See Table1.R

Usage

```
dalycare = load_dalycare_dx()
```

load_bmi_SP

Description

Loads body mass index and body surface area data as defined by BMI()

Usage

```
bmi = load_bmi_SP()
```

load_blood_culture_SP

Description

Loads blood cultures from SP_AlleProevesvar

Usage

```
BC = load_blood_culture_SP()
```

load_common_biochemistry

Description

Loads a list of datasets containing common biochemistry from Lab_forsker.
NB! Remember to save data as suggested in usage.
`labs` = NULL returns options

```
HGB LEU LYM NEU EOS TRC MCV RET FOL B12 JERN TF FER LDH HAP BIL METH KREA CAR B2M URAT NAT KAL FOS CA CA2 PTH DVIT BASP ALAT ASAT DDIM APTT INR
FIBR ADAM KOL HDL LDL TNI TNT TSH TYR HBA1C IGG IGA IGM IGG_MSPIKE IGA_MSPIKE IGM_MSPIKE IGD_MSPIKE FLC MSPIKE U_MSPIKE CRP PCT PSA MTX ANA ANCA
```

You may also specify sets of biochemistry by specifying `labs` as:

```
HEMATOLOGY ANEMIA RENAL KIDNEY HEPATIC LIVER CARDIAC HEART DIC CALCIUM INFECTION
```

Usage

```
LAB_LIST = load_common_biochemistry(labs = c("B2M", "LDH"))
LAB_DF = load_common_biochemistry(labs = c("B2M", "LDH"), combine = TRUE)
```

load_npu_common

Description

Loads a list of vectors containing common NPU codes to your Global Environment.
You may also 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_data() subset
load_data("SDS_lab_forsker", c(NPU.B2M, NPU.LYM), "analysiscode")
View(SDS_lab_forsker_subset)
```

go_live

Description

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

Usage

```
go_live()
```

Definitions

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_data("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_data("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_data("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_data("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.

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

```
SDS_lab_AE = SDS_lab_forsker %>% CTCAE_lab()
```

TX_group
Description

Groups treatment protocols into meaningful groups as class characters.

Usage

```
SP_Behandlingsplaner_del1 %>% TX_group(protocol)
```

filter_virus
Description

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

Usage

```
SP_Bloddyrkning_del1 %>% filter_virus()
```

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; also 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)

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.
Specifying CLL_include = FALSE omits the DC911 score.

Usage

SDS_t_adm %>% CCI()
diagnosis_all %>% CCI(patientid = patientid, icd10 = diagnosis)

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

ATC_AB

Description

Subsets and groups all antimicrobials.

Usage

SDS_epikur %>% ATC_AB()
SP_Administreret_Medicin %>% 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 replaces GCS < 15.

Usage

SP_VitaleVaerdier %>% qSOFA()

BMI
Description

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

Usage

SP_VitaleVaerdier %>% BMI()

BSA
Description

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

Usage

SP_VitaleVaerdier %>% BSA()

CLL_IPI
Description

Calculates CLL-IPI risk as class factor

Usage

RKKP_CLL_CLEAN %>% CLL_IPI()

Citation

da Cunha-Bang et al. *Blood*. Oct 27 2016;128(17):2181-2183.

CLL_WONT
Description

Calculates CLL-WONT risk as class factor

Usage

RKKP_CLL_CLEAN %>% CLL_WONT()

Citation

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

NCCN_IPI
Description

Calculates NCCN-IPI risk for DLBCL as class factor.
NB! Input is complex and generalizable. Use RKKP_LYFO_CLEAN only.

Usage

RKKP_LYFO_CLEAN %>% NCCN_IPI()

Citation

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

MIPI
Description

Calculates MIPI risk for Mantle cell lymphoma as class factor

Usage

RKKP_LYFO_CLEAN %>% MIPI()

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 %>% IPS()

Citation

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

rIPSSWM

Description

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

Usage

RKKP_LYFO_CLEAN %>% rIPSSWM()

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()

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()

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()

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()

House keeping

is_odd
Description

Logical output from numeric values .

Usage

sample(1:10, 5) %>% is_odd()

as_Date
Description

Date output from characters expecting format "%Y-%m-%d".

Usage

"2023-10-17" %>% as_Date()

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

SP_Behandlingsplaner_del1 %>% filter_str_detect(protocol, c("Bendamustin", "Fludara"))

str_between
Description

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

Usage

SP_Journalnotater_del1 %>%
filter(notat_type=="AOP") %>%
mutate(sex = str_between(notat_text, "årig ", c("henvist|møder|kendt"))) %>%
head(10) %>%
pull(text)

```
[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_med, 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 %>% (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

Data %>% n_patients()

nrow_npatients

Description

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

Usage

Data %>% nrow_npatients()

ggplots

KM_plot
Description

Depends on library("ggplot") and library("survminer").
Plots survminer::ggsurvplot with really nice aesthetics.

Usage

fit = survfit(Surv(time, status) ~ CLL_IPI, data)
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

pairwise_survdiff(Surv(time, status) ~ CLL_IPI, data, p.adjust.method = 'none') %>%
tile_pairwise_survdiff(position = 'LL', palette = c(1,2,3,4), labs = FALSE)