

Binary Clinical Scenarios and O2 Device Classifications

Steph Reynolds (Stephanie.Reynolds@ucsf.edu)

Dec 14, 2021 14:05 PM

Background

Project

MINDSCAPE: Modeling of infectious network dynamics for surveillance, control and prevention enhancement

Description

This file imports flowsheet and med admin data and returns a dataset containing all binary clinical scenarios and O2 devices per patient for each day of stay in hospital.

Source Data

- Flowsheet Vitals and Device Data (`flowsheet data table 11.08.2021.csv`)
 - This file contains data on SpO2, FiO2, O2 Flow Rate, O2 Device, Ventilation Modes and Settings, Dialysis Settings, ECMO Settings
- Medical Administration Record (`med admin table 11.08.2021`)
 - This file contains data on Vasopressor (VP) Name, VP Rate, and VP Action.
- Demographics and Events Table (`dm_covid.csv`)
 - This file contains data on patient demographics, admit date, death date (if applicable), and length of stay (LOS).

Load required packages

```
library(here)
library(tidyverse)
library(Hmisc) # this package is only used for "%nin%"
```

Import data

Clean dm table (reformat and select necessary variables)

```
dm <- dm %>%
  select(ID, age, sex, zip, race, ethnicity, smoking, BMI, end_in_death, death_date, LOS) %>%
  group_by(ID) %>%
  distinct()
# can edit select() to include/exclude variables from `dm_raw`
# one row per unique patient ID
```

Clean med table (reformat and select necessary variables)

```
med <- med %>%
  rename(ID = deid_enc_id,      # rename variable names
         time = taken_time) %>%
  mutate(date = as.Date(time, format = "%Y-%m-%d"), # create new column `date` from `taken_time`
         infusion_rate = as.numeric(infusion_rate)) %>% # transform infusion rate to numeric
  select(ID, time, date, vasopressor_flag, infusion_rate) # keep only these variables

# Only include encounter IDs that are included in `dm` table
med <- med[med$ID %in% dm$ID, ]
```

Clean fs table (reformat and select necessary variables)

Note: meas_value is numeric type, and meas_val_chr is character

```
fs <- fs %>%
  rename(ID = deid_enc_id,      # rename variable names
         meas_name = flo_meas_name,
         time = recorded_time) %>%
  filter(clinical_concept %in% c("Respirations", "Urine_output", "Blood_pressure", "Pulse", "Temperature"))
  filter(!is.na(meas_value)) %>% # filter out missing/NA meas_value
  mutate(date = as.Date(time, format = "%Y-%m-%d"), # create new col `date` from `recorded_time`
         meas_val_chr = meas_value,                # create new var `meas_val_chr` to keep meas_value
         meas_value = as.numeric(meas_value)) %>% # transform `meas_value` to numeric type
  select(ID, meas_name, meas_value, meas_val_chr, time, date, clinical_concept) # keep only these variables

# Only include encounter IDs that are included in `dm` table
fs <- fs[fs$ID %in% dm$ID, ]

# n_distinct(fs$ID) # there are n=1,117 unique IDs in `fs` which matches that of `dm`
```

Create table to indicate whether each patient received vasopressor each day

VP=1 if vasopressor_flag = 'Yes' & infusion_rate > 0

```
vp.tab <- med %>%
  mutate(vp_yn = if_else((vasopressor_flag == "Yes" & infusion_rate > 0), 1, 0)) %>% # create new variable `vp_yn`
  group_by(ID, date) %>% # group by ID and date
  mutate(VP = max(vp_yn)) %>% # if pt was on VP at any timestamp during day, then VP = 1 (yes)
  distinct(ID, date, VP) # remove duplicates
```

Create table to indicate whether each patient received ECMO each day

ECMO=1 if clinical_concept == ECMO_settings

```
ecmo.tab <- fs %>%
  mutate(ecmo_yn = if_else(clinical_concept == "ECMO_settings", 1, 0)) %>% # create new variable `ecmo_yn`
  group_by(ID, date) %>% # group by ID and date
  mutate(ECMO = max(ecmo_yn)) %>% # if pt was on ECMO at any time during day, then ECMO = 1 (yes)
  distinct(ID, date, ECMO) # remove duplicates
```

Create table to indicate whether patient received CRRT each day

CRRT=1 if clinical_concept == HD_UF_CRRT_settings and meas_name %in% c("R UFR TRANSCRIBED CRRT IP_CD_UCSF", "R CRRT BLOOD FLOW RATE")

```
crrt.tab <- fs %>%
  mutate(crrt_yn = if_else((clinical_concept == "HD_UF_CRRT_settings" & # create new variable `crrt_yn`
                           meas_name %in% c("R UFR TRANSCRIBED CRRT IP_CD_UCSF",
                                             "R CRRT BLOOD FLOW RATE")), 1, 0)) %>%
  group_by(ID, date) %>% # group by ID and date
  mutate(CRRT = max(crrt_yn)) %>% # if pt received CRRT at any time during day, then CRRT = 1 (yes)
  distinct(ID, date, CRRT) # remove dups
```

Create table to indicate whether patient received support from NIV device each day

NIV=1 if received support from NIV device NIV_per_day: # of times that NIV device was recorded that day

```
niv.tab <- fs %>%
  mutate(niv_yn = if_else(clinical_concept == "NIV_settings", 1, 0)) %>%
  group_by(ID, date) %>%
  mutate(NIV = max(niv_yn),
         NIV_per_day = sum(clinical_concept == "NIV_settings")) %>%
  distinct(ID, date, NIV, NIV_per_day)
```

Create table to indicate whether patient received HD each day

HD=1 if received HD device / settings

```
hd.tab <- fs %>%
  mutate(hd_yn = if_else((clinical_concept == "HD_UF_CRRT_settings" &
    meas_name %in% c("R HD BLOOD FLOW RATE", "R HD ULTRAFILTRATION RATE")), 1, 0)) %>%
  group_by(ID, date) %>%
  mutate(HD = max(hd_yn)) %>%
  distinct(ID, date, HD)
```

Create table to indicate whether patient was intubated each day

INTUB=1 if received intubation

```
intub.tab <- fs %>%
  mutate(intub_yn = if_else(clinical_concept == "Intubation_settings", 1, 0)) %>%
  group_by(ID, date) %>%
  mutate(INTUB = max(intub_yn)) %>%
  distinct(ID, date, INTUB)
```

Filter data for SpO2 and FiO2 and merge tables to calculate SF

SF = SpO2 / FiO2 * 100

```
# Filter data for SpO2 values and assign to table `spo2.tab`
spo2.tab <- fs %>%
  filter(clinical_concept == "SpO2") %>%
  select(ID, time, date, meas_value)

# Filter data for FiO2 values and assign to table `fio2.tab`
fio2.tab <- fs %>%
  filter(meas_name == "R FIO2") %>%
  select(ID, time, date, meas_value)

# Merge SpO2 and FiO2 tables and calculate SF
SF.tab <- fio2.tab %>%
  left_join(spo2.tab, by = c("ID", "time"), suffix = c(".fi", ".sp")) %>%
  mutate(SF = (meas_value.sp / meas_value.fi * 100)) %>%
  select(ID, time, date.fi, SF) %>%
  rename(date = date.fi)
```

Create table to indicate whether patient experienced SpO2/FiO2<200 at same timestamp, and assign to new variable SF_LT_200

SF_LT_200=1 if SpO2/FiO2<200 at same timestamp

```
SF200.tab <- SF.tab %>%
  mutate(sf_lt_200_yn = if_else(SF < 200, 1, 0)) %>%
  group_by(ID, date) %>%
  mutate(SF_LT_200 = max(sf_lt_200_yn)) %>%
  distinct(ID, date, SF_LT_200)
```

Assign all O2 devices to respiratory support categories: NONE, SIMPLE, NIV, IV, CPAP, and NC (nasal cannula and high-flow nasal cannula)

```
# Read in `o2_dev_names.csv` which lists all O2 devices and their respective categories
o2_dev_names <- read_csv(here("data", "o2_dev_names.csv"))
```

```
## Rows: 212 Columns: 3
```

```
## -- Column specification -----
## Delimiter: ","
## chr (2): dev_names, dev_cat
## dbl (1): nc_dev
```

```
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
# NONE -- filter all rows where dev_cat == "no_dev"
no_dev_names <- filter(o2_dev_names, dev_cat == "no_dev")[[1]]

# SIMPLE -- filter all rows where dev_cat == "simple_dev"
simple_dev_names <- filter(o2_dev_names, dev_cat == "simple_dev")[[1]]

# Non-Invasive Ventilation (NIV) -- filter all rows where dev_cat == "NIV_dev"
niv_dev_names <- filter(o2_dev_names, dev_cat == "NIV_dev")[[1]]

# Invasive (IV) -- filter all rows where dev_cat == "iv_dev"
iv_dev_names <- filter(o2_dev_names, dev_cat == "iv_dev")[[1]]

# CPAP -- filter all rows where dev_cat == "cpap_dev"
cpap_dev_names <- filter(o2_dev_names, dev_cat == "cpap_dev")[[1]]

# Nasal cannula or high-flow nasal cannula (HFNC) -- filter all rows where nc_dev == 1
nc_dev_names <- filter(o2_dev_names, nc_dev == 1)[[1]]
```

Create table of O2 devices where each column/variable is a different category of respiratory support devices, indicating whether patient received respiratory support and if so, from which category (NONE, SIMPLE, NIV, IV, CPAP, NC)

```
device.tab <- fs %>%
  filter(clinical_concept == "O2_device") %>%
  mutate(NOdev_yn = if_else(meas_val_chr %in% no_dev_names, 1, 0),
         SIMPLEdev_yn = if_else(meas_val_chr %in% simple_dev_names, 1, 0),
         NIVdev_yn = if_else(meas_val_chr %in% niv_dev_names, 1, 0),
         IVdev_yn = if_else(meas_val_chr %in% iv_dev_names, 1, 0),
         CPAPdev_yn = if_else(meas_val_chr %in% cpap_dev_names, 1, 0),
         NCdev_yn = if_else(meas_val_chr %in% nc_dev_names, 1, 0)) %>%
  group_by(ID, date) %>%
  mutate(NODEV = max(NOdev_yn),
         SIMPLEDEV = max(SIMPLEdev_yn),
         SIMPLE_PER_DAY = sum(meas_val_chr %in% simple_dev_names),
         NIVDEV = max(NIVdev_yn),
         IVDEV = max(IVdev_yn),
         CPAPDEV = max(CPAPdev_yn),
         NCDEV = max(NCdev_yn),
         NC_PER_DAY = sum(meas_val_chr %in% nc_dev_names)) %>%
  distinct(ID, date, NODEV, SIMPLEDEV, SIMPLE_PER_DAY, NIVDEV, IVDEV, CPAPDEV, NCDEV, NC_PER_DAY)
```

Create table to indicate whether patients had O2 use (O2) and if so, low (LowO2) or high O2 flow rate (HighO2)

O2: meas_value > 0 then O2_use = 1 (yes) LowO2: meas_value <= 12 HighO2: meas_value > 12

```
# Create vector for O2 flow rate names
O2_names <- c("R OXYGEN FLOW RATE", "R OXYGEN FLOW RATE 2 IP_CD_UCSF")

# Create table with columns 'O2', 'LowO2', and 'HighO2'
O2.tab <- fs %>%
  mutate(O2 = if_else(meas_name %in% O2_names & meas_value > 0, 1, 0),
         LowO2 = if_else(meas_name %in% O2_names & meas_value <= 12, 1, 0),
         HighO2 = if_else(meas_name %in% O2_names & meas_value > 12, 1, 0)) %>%
  group_by(ID, date) %>%
  mutate(O2 = max(O2),
         LowO2 = max(LowO2),
         HighO2 = max(HighO2)) %>%
  distinct(ID, date, O2, LowO2, HighO2)
```

Merge device.tab and O2.tab to create O2_DEV.tab

```
# Left_join on 'ID' and 'date' columns
O2_DEV.tab <- O2.tab %>%
  left_join(device.tab, by = c("ID", "date")) %>%           #left join????
  mutate(NC_GT_12 = if_else(NCDEV == 1 & HighO2 == 1, 1, 0))
```

Merge all tables to aggregate binary clinical scenarios and O2 device classifications

List of tables to join include: - fs - dm - vp.tab - ecmo.tab - crrt.tab - niv.tab - hd.tab - intub.tab - SF200.tab - O2_DEV.tab

```
bin_clin_scen_df <- fs %>%
  distinct(ID, date) %>%
  left_join(dm, by = "ID", suffix = c("", ".dupcol")) %>%
  left_join(vp.tab, by = c("ID", "date"), suffix = c("", ".dupcol")) %>%
  left_join(ecmo.tab, by = c("ID", "date"), suffix = c("", ".dupcol")) %>%
  left_join(crrt.tab, by = c("ID", "date"), suffix = c("", ".dupcol")) %>%
  left_join(niv.tab, by = c("ID", "date"), suffix = c("", ".dupcol")) %>%
  left_join(hd.tab, by = c("ID", "date"), suffix = c("", ".dupcol")) %>%
  left_join(intub.tab, by = c("ID", "date"), suffix = c("", ".dupcol")) %>%
  left_join(SF200.tab, by = c("ID", "date"), suffix = c("", ".dupcol")) %>%
  left_join(O2_DEV.tab, by = c("ID", "date"), suffix = c("", ".dupcol")) %>%
  select(-ends_with(".dupcol")) %>%
  mutate(DEATH = 0,
         DEATH = if_else(date == death_date, 1, 0))
```

Save resulting df and export as .csv

```
write_csv(bin_clin_scen_df, here("data", "binary_clin_scen_df_11.08.21.csv"))
save(bin_clin_scen_df, file = here("data", "binary_clin_scen_df_11.08.21.Rdata"))
```

End of Document