Missing Data Tutorial

Yichi Zhang

2023-03-30

```
# import the linelist
linelist <- import("linelist_cleaned.rds")
# view the first 6 observations of the dataset
head(linelist)
```

```
##   case_id generation date_infection date_onset date_hospitalisation
## 1  5fe599          4     2014-05-08 2014-05-13           2014-05-15
## 2  8689b7          4           <NA> 2014-05-13           2014-05-14
## 3  11f8ea          2           <NA> 2014-05-16           2014-05-18
## 4  b8812a          3     2014-05-04 2014-05-18           2014-05-20
## 5  893f25          3     2014-05-18 2014-05-21           2014-05-22
## 6  be99c8          3     2014-05-03 2014-05-22           2014-05-23
##   date_outcome outcome gender age age_unit age_years age_cat age_cat5
## 1         <NA>    <NA>      m   2    years         2     0-4      0-4
## 2   2014-05-18 Recover      f   3    years         3     0-4      0-4
## 3   2014-05-30 Recover      m  56    years        56   50-69    55-59
## 4         <NA>    <NA>      f  18    years        18   15-19    15-19
## 5   2014-05-29 Recover      m   3    years         3     0-4      0-4
## 6   2014-05-24 Recover      f  16    years        16   15-19    15-19
##                              hospital       lon      lat infector source wt_kg
## 1                               Other -13.21574 8.468973   f547d6  other    27
## 2                             Missing -13.21523 8.451719     <NA>   <NA>    25
## 3 St. Mark's Maternity Hospital (SMMH) -13.21291 8.464817     <NA>   <NA>    91
## 4                        Port Hospital -13.23637 8.475476   f90f5f  other    41
## 5                    Military Hospital -13.22286 8.460824   11f8ea  other    36
## 6                        Port Hospital -13.22263 8.461831   aec8ec  other    56
##   ht_cm ct_blood fever chills cough aches vomit temp time_admission       bmi
## 1    48       22    no     no   yes    no   yes 36.8           <NA> 117.18750
## 2    59       22  <NA>   <NA>  <NA>  <NA>  <NA> 36.9          09:36  71.81844
## 3   238       21  <NA>   <NA>  <NA>  <NA>  <NA> 36.9          16:48  16.06525
## 4   135       23    no     no    no    no    no 36.8          11:22  22.49657
## 5    71       23    no     no   yes    no   yes 36.9          12:60  71.41440
## 6   116       21    no     no   yes    no   yes 37.6          14:13  41.61712
##   days_onset_hosp
## 1               2
## 2               1
## 3               2
## 4               2
## 5               1
## 6               1
```

Problems with Missingness

Sometimes you might encounter the following errors when you started analyzing your data...

```
mean(linelist$age)
```

```
## [1] NA
```

These functions could work after dealing with missing data. In R, NA represents all types of missing data.

```
mean(linelist$age, na.rm = TRUE)
# use na.omit to exclude incomplete cases, sample size will change
mod_omit <- lm(bmi ~ age + gender + wt_kg + ht_cm, data = linelist, na.action = na.omit)
resid(mod_omit)
# use na.exclude to exclude incomplete cases in the analysis but still keep the them in the dataset
mod_exclude <- lm(bmi ~ age + gender + wt_kg + ht_cm, data = linelist, na.action = na.exclude)
resid(mod_exclude)
# could check the help page of na.action to see how R handles missing values
# ? na.action
```

Assess Missingness

The number of NAs in the dataset by columns.

```
# check the number of NAs in the dataset by columns
colSums(is.na(linelist))
```

```
##              case_id           generation       date_infection
##                    0                    0                 2087
##           date_onset date_hospitalisation         date_outcome
##                  256                    0                  936
##              outcome               gender                  age
##                 1323                  278                   86
##             age_unit            age_years              age_cat
##                    0                   86                   86
##             age_cat5             hospital                  lon
##                   86                    0                    0
##                  lat             infector               source
##                    0                 2088                 2088
##               wt_kg                ht_cm            ct_blood
##                    0                    0                    0
##               fever               chills                cough
##                  249                  249                  249
##               aches               vomit                 temp
##                  249                  249                  149
##       time_admission                  bmi      days_onset_hosp
##                  765                    0                  256
```

```
# the dimension of the original dataset
dim(linelist)
```

```
## [1] 5888   30
```

The proportion of missing values in each variable.

```
linelist %>%
    # check each variable's missing values
    map(is.na) %>%
    # calculate the total sum of missing values in each variable
```

```
    map(sum) %>%
    # pick the sum of missing values in each variable and divide by the sample size
    map(~ . /nrow(linelist))%>%
    # bind multiple columns together
    bind_cols()
```

```
## # A tibble: 1 x 30
##   case_id genera~1 date_~2 date_~3 date_~4 date_~5 outcome gender    age age_u~6
##     <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>   <dbl>
## 1       0        0   0.354  0.0435       0   0.159   0.225 0.0472 0.0146       0
## # ... with 20 more variables: age_years <dbl>, age_cat <dbl>, age_cat5 <dbl>,
## #   hospital <dbl>, lon <dbl>, lat <dbl>, infector <dbl>, source <dbl>,
## #   wt_kg <dbl>, ht_cm <dbl>, ct_blood <dbl>, fever <dbl>, chills <dbl>,
## #   cough <dbl>, aches <dbl>, vomit <dbl>, temp <dbl>, time_admission <dbl>,
## #   bmi <dbl>, days_onset_hosp <dbl>, and abbreviated variable names
## #   1: generation, 2: date_infection, 3: date_onset, 4: date_hospitalisation,
## #   5: date_outcome, 6: age_unit
```

We can compute missingness in different ways.

```
# percent of ALL data frame values that are missing
pct_miss(linelist)
```

```
## [1] 6.688745
```

```
# percent of rows with any value missing
pct_miss_case(linelist)
```

```
## [1] 69.12364
```

```
# percent of rows that are complete
pct_complete_case(linelist)
```

```
## [1] 30.87636
```

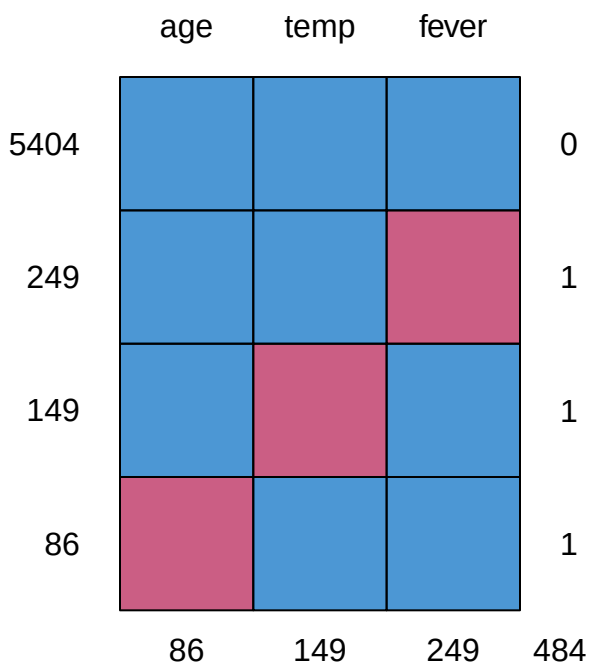
Functions to Remove Missing Values

```
## listwise deletion
na.omit(linelist)
## drops rows missing values for any of these columns
linelist %>%
  drop_na(case_id, date_onset, age)
## remove NAs by specifing the input parameters
mean(linelist$age, na.rm = TRUE)
## complete cases
linelist[complete.cases(linelist), ]
```

Visualize Missingness

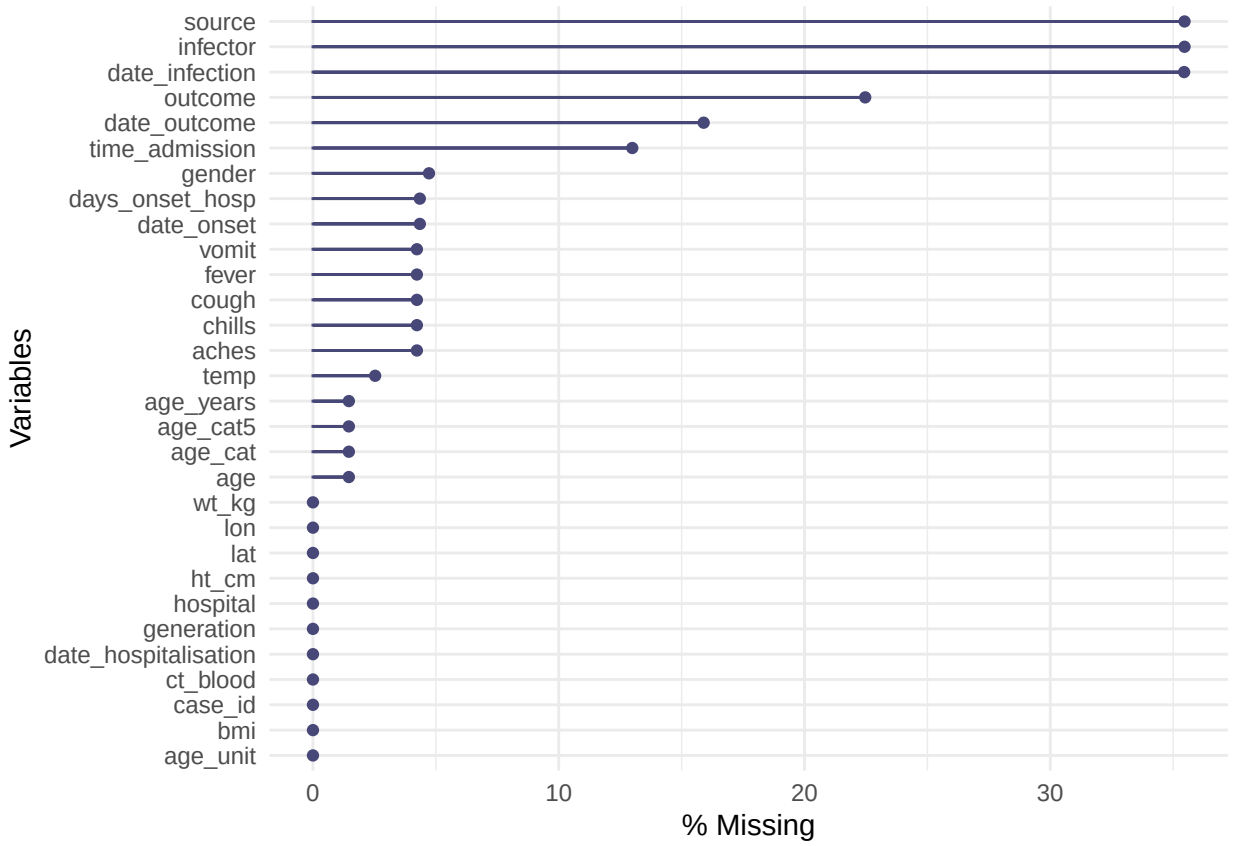
We can use `vis_miss()` to visualize the dataframe as a heatmap, showing whether each value is missing or not.

```
# check missing data pattern for variables age, temperature, and fever
md.pattern(linelist[, c("age", "temp", "fever")])
```

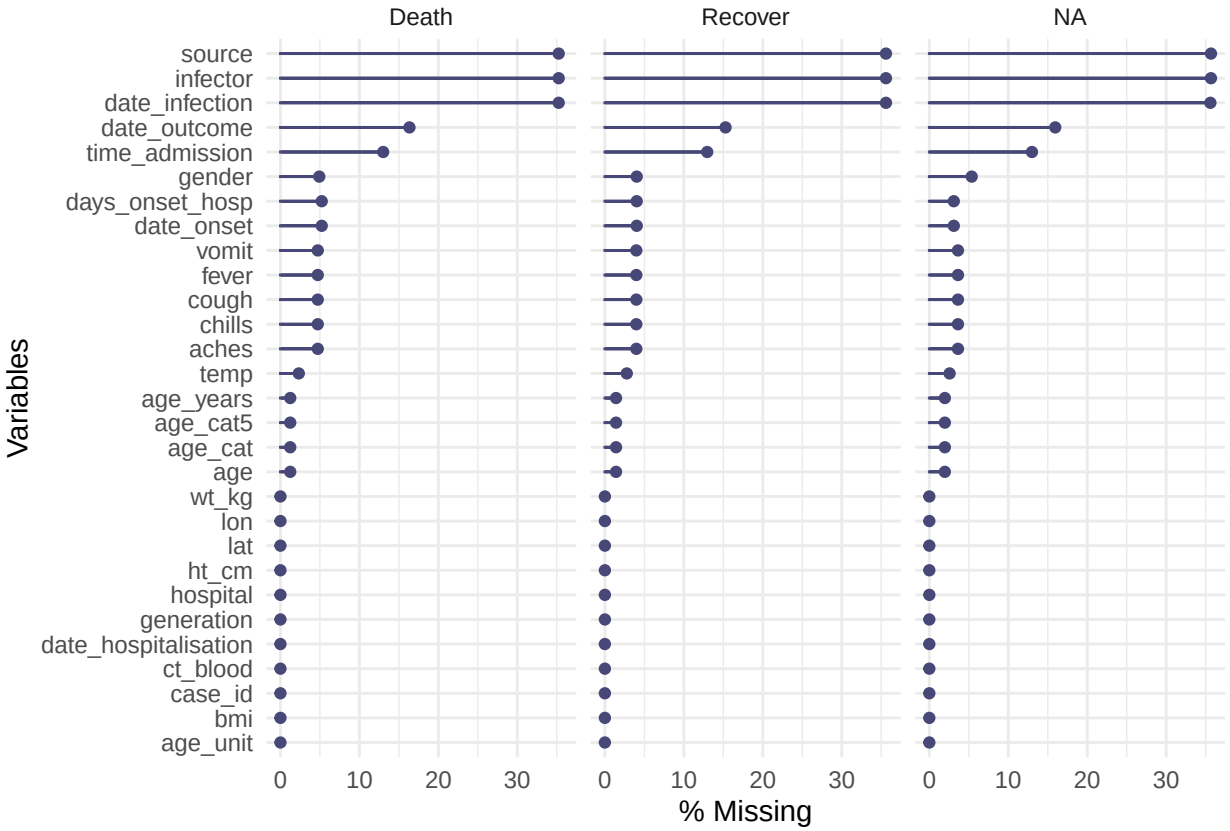



```
##      age temp fever
## 5404   1    1     1   0
## 249    1    1     0   1
## 149    1    0     1   1
## 86     0    1     1   1
##       86  149   249 484
```

```
## show the number of missing in each column
gg_miss_var(linelist, show_pct = TRUE)
```

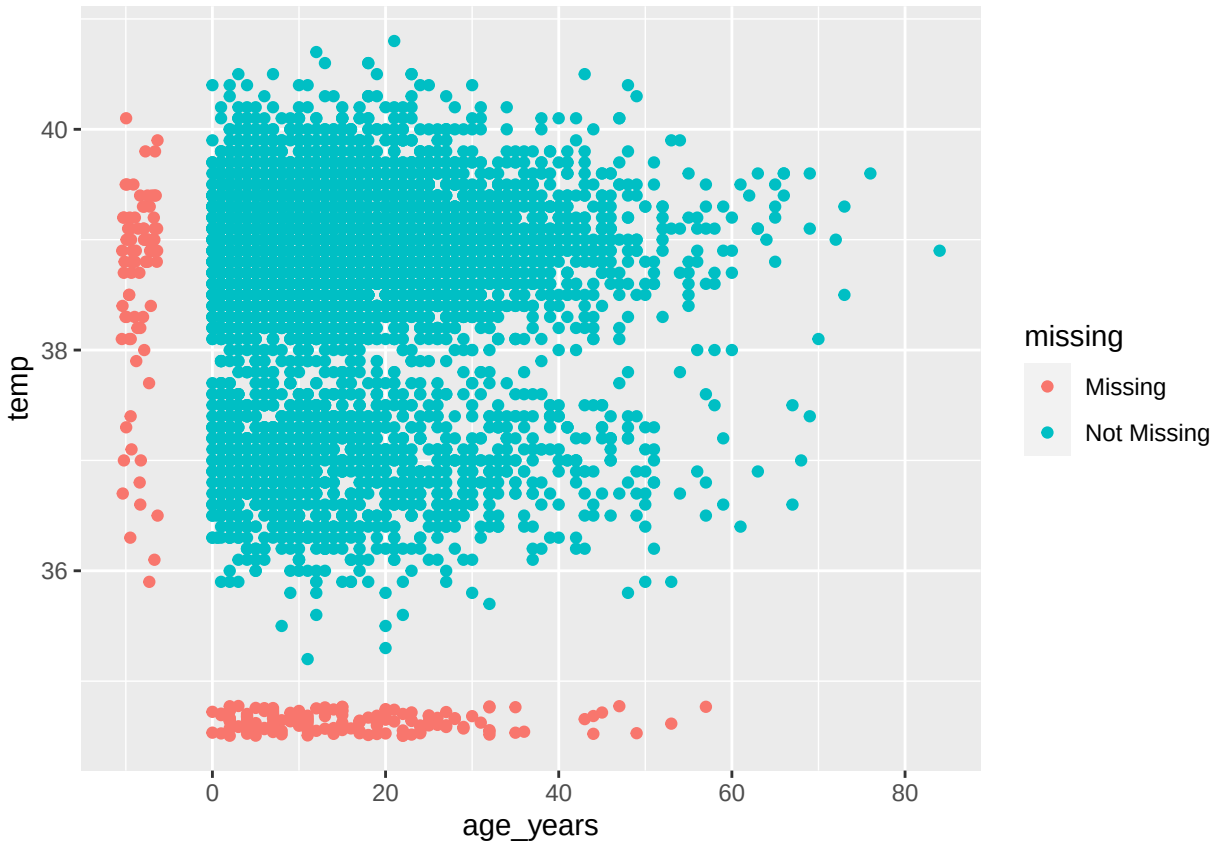



```
## split the data by a variable
linelist %>%
  gg_miss_var(show_pct = TRUE, facet = outcome)
```



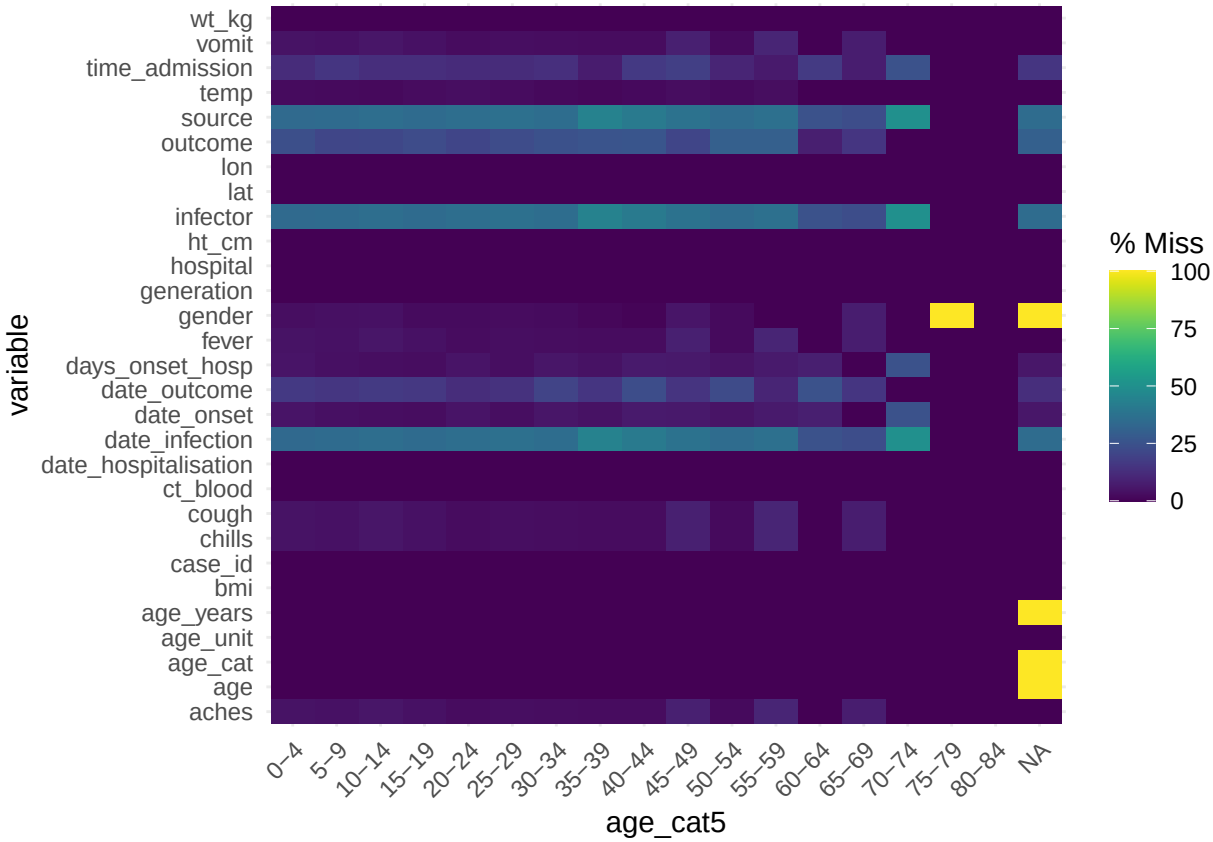
In the scatterplot below, the red dots are records where the value for one column is present but the value for the other column is missing. This allows you to see the distribution of missing values in relation to the non-missing values.

```
ggplot(
  data = linelist,
  mapping = aes(x = age_years, y = temp)) +
  geom_miss_point()
```



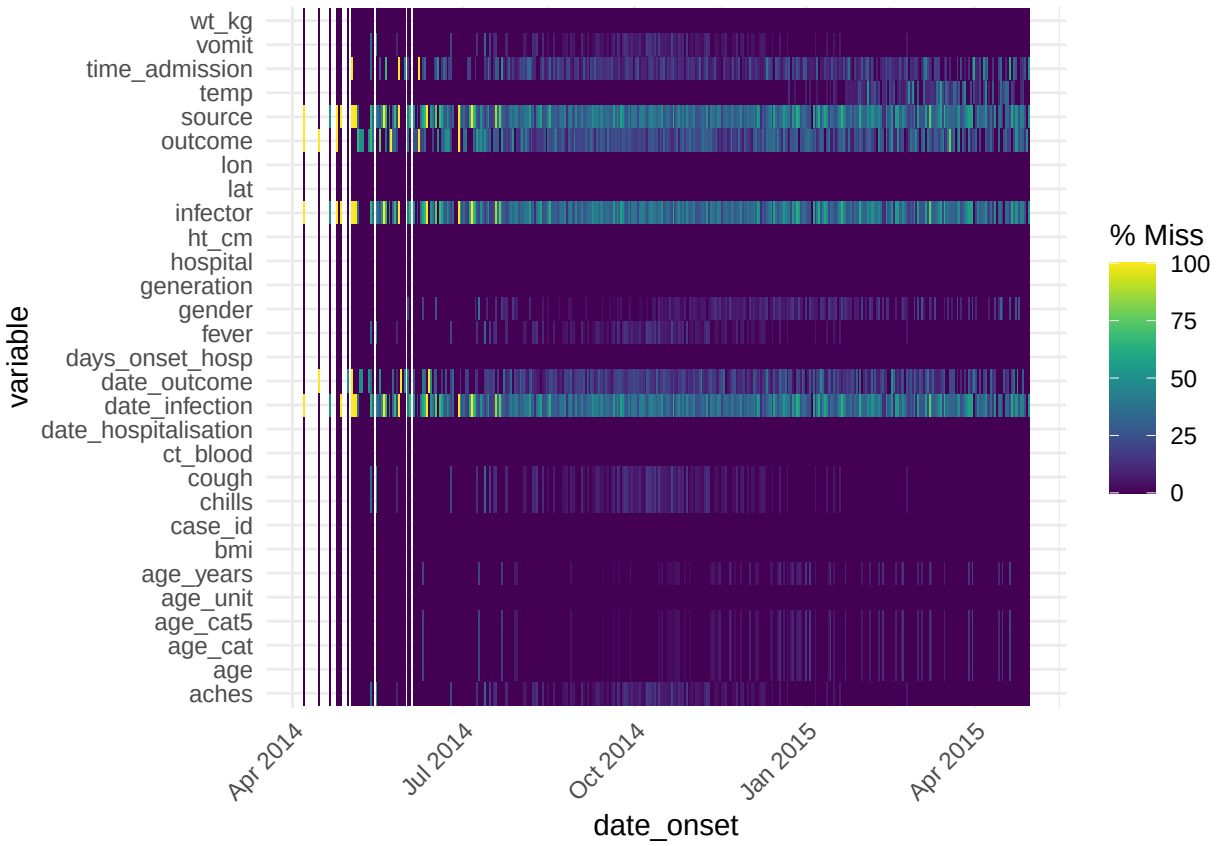
`gg_miss_fct()` assesses missingness in the data frame stratified by another column, which returns a heatmap of percent missingness in the data frame by a factor/categorical (or date) column.

```
gg_miss_fct(linelist, age_cat5)
```




```
## change over time
gg_miss_fct(linelist, date_onset)
```

```
## Warning: Removed 29 rows containing missing values (‘geom_tile()‘).
```



Here we take the linelist, add a new column for week, group the data by week, and then calculate the percent of that week's records where the value is missing. Then we plot the proportion missing as a line, by week.

```
outcome_missing <- linelist %>%
    mutate(week = lubridate::floor_date(date_onset, "week")) %>% # create new week column
    group_by(week) %>%                                           # group the rows by week
    summarise(                                                   # summarize each week
        n_obs = n(),                                             # number of records
        outcome_missing = sum(is.na(outcome) | outcome == ""),   # number of records missing the value
        outcome_p_miss  = outcome_missing / n_obs,               # proportion of records missing the va
        outcome_dead    = sum(outcome == "Death", na.rm=T),      # number of records as dead
        outcome_p_dead  = outcome_dead / n_obs) %>%              # proportion of records as dead
    tidyr::pivot_longer(-week, names_to = "statistic") %>%       # pivot all columns except week, to lo
    filter(stringr::str_detect(statistic, "_p_"))                # keep only the proportion values

ggplot(data = outcome_missing)+
    geom_line(
      mapping = aes(x = week, y = value, group = statistic, color = statistic),
      size = 2,
      stat = "identity")+
    labs(title = "Weekly outcomes",
         x = "Week",
         y = "Proportion of weekly records") +
     scale_color_discrete(
       name = "",
       labels = c("Died", "Missing outcome"))+
    scale_y_continuous(breaks = c(seq(0,1,0.1)))+
```

```
  theme_minimal()+
  theme(legend.position = "bottom")
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
```

```
## Warning: Removed 2 rows containing missing values ('geom_line()').
```

Address Missingness

Mean Imputation

```
linelist[is.na(linelist$temp),]
linelist <- linelist %>%
    mutate(temp_replace_na_with_mean = replace_na(temp, mean(temp, na.rm = T)))
linelist[is.na(linelist$temp),]
```

Regression Imputation

```
simple_temperature_model_fit <- lm(temp ~ fever + age_years, data = linelist)
#using our simple temperature model to predict values just for the observations where temp is missing
predictions_for_missing_temps <- predict(simple_temperature_model_fit,
                                         newdata = linelist %>% filter(is.na(temp)))
```

```
model_dataset <- linelist %>%
    select(temp, fever, age_years)
temp_imputed <- mice(model_dataset,
                     # linear regression, predicted values
```

```
                      method = "norm.predict",
                      # set seed for reproducibility
                      seed = 1,
                      # number of multiple imputations, default 5
                      m = 1,
                      print = F)
```

```
## Warning: Number of logged events: 1
```

Multiple Imputation

```
# imputing missing values for all variables in our model_dataset, and creating 10 new imputed datasets
multiple_imputation = mice(
    model_dataset,
    seed = 1,
    m = 10,
    print = FALSE)
```

```
## Warning: Number of logged events: 1
```

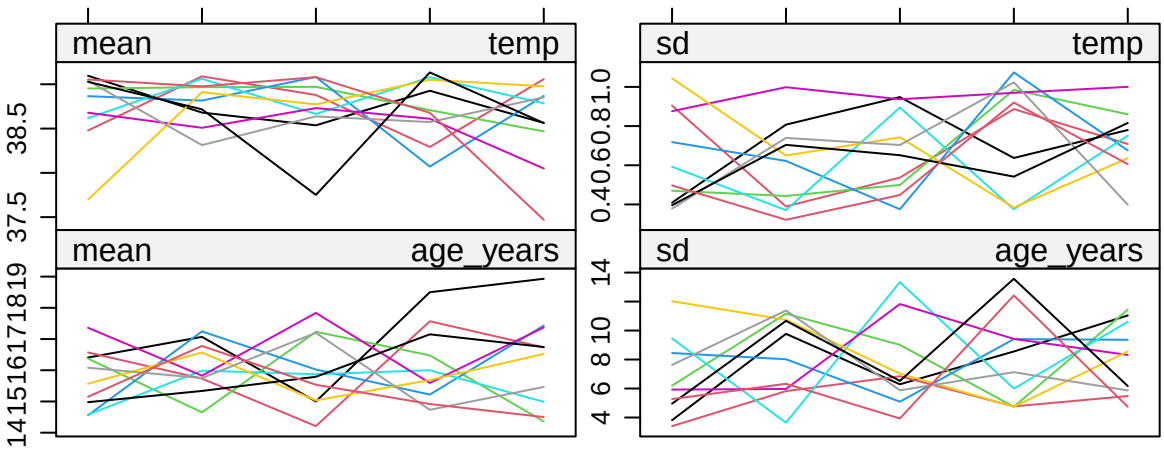
```
# inspect the regression model with the imputed data
model_fit <- with(multiple_imputation, lm(temp ~ age_years + fever))
base::summary(mice::pool(model_fit))
```

```
##          term     estimate    std.error    statistic        df       p.value
## 1 (Intercept) 3.703143e+01 0.0270863456 1.367162e+03  26.83673  1.583113e-66
## 2   age_years 3.867829e-05 0.0006090202 6.350905e-02 171.44363  9.494351e-01
## 3    feveryes 1.978044e+00 0.0193587115 1.021785e+02 176.51325 5.666771e-159
```

```
pool(model_fit)
```

```
## Class: mipo    m = 10
##          term  m     estimate         ubar            b            t dfcom
## 1 (Intercept) 10 3.703143e+01 3.111913e-04 3.840716e-04 7.336701e-04  5636
## 2   age_years 10 3.867829e-05 2.876084e-07 7.572478e-08 3.709056e-07  5636
## 3    feveryes 10 1.978044e+00 2.918569e-04 7.536619e-05 3.747597e-04  5636
##          df       riv    lambda       fmi
## 1  26.83673 1.3576175 0.5758430 0.6042749
## 2 171.44363 0.2896204 0.2245780 0.2334683
## 3 176.51325 0.2840529 0.2212159 0.2298925
```

```
plot(multiple_imputation)
```



The fraction of information missing due to nonresponse (fmi) and the relative increase in variance due to nonresponse (lambda) are pretty high.

```
## imputed datasets
multiple_imputation
```

```
## Class: mids
## Number of multiple imputations:  10
## Imputation methods:
##      temp     fever age_years
##     "pmm"        ""     "pmm"
## PredictorMatrix:
##           temp fever age_years
## temp         0     0         1
## fever        0     0         0
## age_years    1     0         0
## Number of logged events:  1
##   it im dep     meth   out
## 1  0  0     constant fever
```

```
# information stores in the object multiple_imputation
attributes(multiple_imputation)
```

```
## $names
##  [1] "data"            "imp"             "m"               "where"
##  [5] "blocks"          "call"            "nmis"            "method"
##  [9] "predictorMatrix" "visitSequence"   "formulas"        "post"
## [13] "blots"           "ignore"          "seed"            "iteration"
## [17] "lastSeedValue"   "chainMean"       "chainVar"        "loggedEvents"
## [21] "version"         "date"
##
## $class
```

```
## [1] "mids"
```

```
# original dataset
# multiple_imputation$data
# imputated dataset
# multiple_imputation$imp
# extract a certain imputed dataset
c3 <- complete(multiple_imputation, 1)
```

Exercise

Explore the following dataset `airquality`

```
airquality
```