This post is a supplementary material for an assignment. The assignment is part of the Augmented Machine Learning unit for a Specialised Diploma in Data Science for Business. The aim of the assignment is to use DataRobot for predictive modelling. Exploratory data analysis and feature engineering will be done here in R before the data is imported into DataRobot

Intro

The aim of this project is to classify if patients with Community Acquired Pneumonia (CAP) became better after seeing a doctor or became worse despite seeing a doctor. The variables of the dataset can be classified into 13 categories. The first 8 categories have been explored in the previous post. The remaining categories will be explored in this post.

```
library(tidyverse)
theme_set(theme_light())

# previously and partial EDA dataset
load("CAP_EDA1.RData")

# 13 categories
categories13< - readxl::read_excel("Incidence rate of community-acquired pneumonia in adults a
population-based prospective active surveillance study in three cities in South America.xls",
sheet=3)

categories13 %>%  DT::datatable(rownames = F, options = list(searchHighlight = TRUE, paging=
T))
```

\n \n

Prefix< \/th>\n < \/tr>\n < \/thead>\n< \/table>","options":{"searchHighlight":true,"paging":true,"columnDefs":[{"className":"dt-right","targets":0}],"order":[],"autoWidth":false,"orderClasses":false}},"evals":[],"jsHooks":[]}

Customized EDA functions from the previous post will be used here.

```
dtype< - function(datafr, x){
datafr%>% select(starts_with(x, ignore.case = F)) %>% str()
}

eda_c< - function(datafr,x){
datafr %>% select(starts_with(x, ignore.case = F)) %>%  map(~ table(.x, useNA = "always"))
}

eda_n_NAplt< - function (datafr, x){
datafr %>% select(starts_with(x, ignore.case = F)) %>% summarise(across(starts_with(x),
~mean(is.na(.)))) %>% pivot_longer(cols = everything(), names_to= "Variables" ,
values_to="pct_na") %>% mutate(Variables= fct_reorder(Variables, pct_na)) %>%
ggplot(aes(x=Variables, y=pct_na, fill= pct_na))+ geom_col() + coord_flip() +
scale_y_continuous(labels=scales::percent_format()) + scale_fill_viridis_c(option = "plasma")}
```

#< \/th>\n Category< \/th>\n Category description< \/th>\n Examples< \/th>\n

```
eda_n_NAcutoff< - function(datafr, x, low, high){
datafr%>% select(starts_with(x, ignore.case = F)) %>% summarise(across(starts_with(x),
~mean(is.na(.)))) %>% pivot_longer(cols = everything(), names_to="Variables",
values_to="pct_na") %>% filter((pct_na>low & pct_na% pull(Variables)}

eda_n_outlier< -function(datafr, x_selected){
# nested df with plots
plt<-datafr %>% select(all_of(x_selected)) %>% pivot_longer(cols=everything()
,names_to="Variables", values_to="values") %>% nest(-Variables) %>% mutate(plot= map2(.x= data,
.y= Variables,
~ggplot(data=.x, aes(x= values)) + geom_boxplot() + labs(title = .y)
))
# print the plots
for (i in 1:length(x_selected)){
p< -plt[[3]][[i]]
print(p)}
}
```

\n

9 Lab_ related category

Sodium levels Lab_Na should be in numeric form but it is registered as a string. Upon closer inspection, there are no characters found in the variable. The variable can be converted into a numeric variable.

```
(dtype(df,"Lab"))
```

```
## tibble [2,112 x 20] (S3: tbl_df/tbl/data.frame)
##  $ Lab_RBC        : num [1:2112] 32 43 27 33 10.4 33 45 35 35.5 21.3 ...
##  $ Lab_Hb         : num [1:2112] 10.7 14.2 9.6 11.4 4.8 11 13.3 11.2 11 6.8 ...
##  $ Lab_WBC        : num [1:2112] 15.7 12.5 6.6 9.5 75.3 18.9 8.3 13.1 8.5 13.9 ...
##  $ Lab_NeuImu     : num [1:2112] NA NA NA NA NA NA NA NA NA NA ...
##  $ Lab_Neu        : num [1:2112] 89 80 86 76 NA 89 88 82 81 92 ...
##  $ Lab_plt        : num [1:2112] 175 170 120 274 27 128 333 621 496 180 ...
##  $ Lab_Na         : chr [1:2112] NA NA NA NA ...
##  $ Lab_urea       : num [1:2112] 60 NA 99 56 143 56.3 49 19 25 214 ...
##  $ Lab_Cr         : num [1:2112] 1.61 NA 0.77 0.84 2.94 0.88 0.95 0.83 0.73 8.11 ...
##  $ Lab_Bicarb     : num [1:2112] NA NA NA NA NA NA NA NA NA NA ...
##  $ Lab_Sugar      : num [1:2112] 76 NA 83 111 88 70 93 78 100 75 ...
##  $ Lab_Alb        : num [1:2112] NA NA NA NA NA NA NA NA NA NA ...
##  $ Lab_lactate    : num [1:2112] NA NA NA NA NA NA NA NA NA NA ...
##  $ Lab_lactateHigh: chr [1:2112] "Unavailable" "Unavailable" "Unavailable" "Unavailable" ...
##  $ Lab_CRP        : num [1:2112] 48 96 NA 92 192 48 96 48 48 192 ...
##  $ Lab_CRPHigh    : chr [1:2112] "Yes" "Yes" "Unavailable" "Yes" ...
##  $ Lab_pH         : num [1:2112] NA NA NA NA NA NA NA NA NA NA ...
##  $ Lab_CO2        : num [1:2112] NA NA NA NA NA NA NA NA NA NA ...
##  $ Lab_O2         : num [1:2112] NA NA NA NA NA NA NA NA NA NA ...
##  $ Lab_FiO2       : num [1:2112] NA NA NA NA NA NA NA NA NA NA ...
```

```
## NULL
```

```
# check for characters
(df %>% mutate(char= str_detect(Lab_Na, pattern = "[A-z0-9]"))  %>% filter(Lab_Na==T))
```

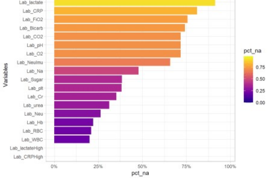
```
## # A tibble: 0 x 131
## # ... with 131 variables: Pt_Site , Pt_CaseNumber , Pt_Age ,
## #   R_CXR_infiltrate , R_CXR_cavitation , R_CXR_effusionSite ,
## #   R_CT_inflitrate , R_CT_cavitation , R_CT_effusionSite ,
## #   SS_cough , SS_phlegm , SS_lungSounds , SS_temp ,
## #   SS_breathing , SS_daysOfRespSymp , Hx_mass ,
## #   Hx_stroke , Hx_kidney , Hx_liver , Hx_brainMental ,
## #   Hx_diabetes , Hx_pastCAP , Hx_asp , Hx_alcohol ,
## #   Hx_immune , Hx_COPD , Social_drugs ,
## #   Social_overcrowded , Hx_heart_type , Social_smoke ,
## #   Hx_HIV , HCAP_hospStay , HCAP_IVAbx , HCAP_Chemo ,
## #   HCAP_diaylsis , HCAP_injury , PE_AMS , PE_HR ,
## #   PE_RR , PE_BP_S , PE_BP_D , PE_temp , Lab_RBC ,
## #   Lab_Hb , Lab_WBC , Lab_NeuImu , Lab_Neu ,
## #   Lab_plt , Lab_Na , Lab_urea , Lab_Cr ,
## #   Lab_Bicarb , Lab_Sugar , Lab_Alb , Lab_lactate ,
## #   Lab_lactateHigh , Lab_CRP , Lab_CRPHigh , Lab_pH ,
## #   Lab_CO2 , Lab_O2 , Lab_FiO2 , CS_Resp , CS_Blood ,
## #   CS_Urine , CS_screen , CS_agent , CS_Organism1 ,
## #   CS_Organism1Blood , CS_Organism1Sputum ,
## #   CS_Organism1Tracheal , CS_Organism1BAL , CS_Organism1Urine ,
## #   CS_Organism1Sero , CS_Organism1Other ,
## #   CS_Organism1Comments , CS_Organism2 , CS_Orgainsim2Blood ,
## #   CS_Organism2Sputum , CS_Organism2Tracheal ,
## #   CS_Organism2BAL , CS_Organism2Urine , CS_OrganismSero ,
## #   CS_OrganismOther , CS_OrganismComments ,
## #   Abx_AmoxicillinSulbactam , Abx_AmoxicillinSulbactamOral ,
## #   Abx_AmoxicillinSulbactamNonoral ,
## #   Abx_AmoxicillinSulbactamNonoralStart ,
## #   Abx_AmoxicillinSulbactamNonoralEnd , Abx_Ampicillin ,
## #   Abx_AmpicillinStart , Abx_AmpicillinEnd ,
## #   Abx_AmpicillinSulbactam , Abx_Azithromycin ,
## #   Abx_Ceftriaxone , Abx_Cefotaxime , Abx_ClarithromycinOral ,
## #   Abx_Cefepime , Abx_CefepimeStart , ...
```

```
# convert Lab_na to num
df< -df %>% mutate(Lab_Na=as.numeric(Lab_Na))
```

Missing Lab values

More than half of the Lab_ variables have >40% missing values. These variables will be removed. Lab_CRPHigh and Lab_lactateHigh are binary variables indicating if CRP Lab_CRP and lactate levels Lab_lactate are above normal limits. As Lab_CRP and Lab_lactate will be dropped due to too many missing values, Lab_CRPHigh and Lab_lactateHigh

```
(eda_n_NAplt(df,"Lab"))
```



```
# remove unwanted Lab_ col
lab_selected< -eda_n_NAcutoff(df, "Lab", 0, .40)
lab_all<-df %>% select(starts_with("Lab")) %>% colnames()
lab_removed< -setdiff(lab_all, lab_selected)
df<- df %>% select(-any_of(lab_removed))
```

Outlier `Lab` values

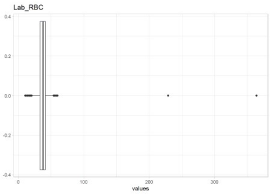
The following `Lab` values have numerous outliers warranting further investigation:

- Haemoglobin levels , `Lab_Hb` >100
- Neutrophil levels , `Lab_Neu`, <35
- Sugar levels, `Lab_Sugar` > 450

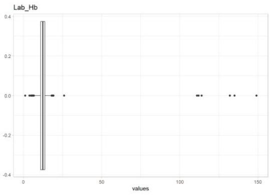
```
eda_n_outlier(df, lab_selected)
```

```
## Warning: All elements of `...` must be named.
## Did you want `data = c(values)`?
```

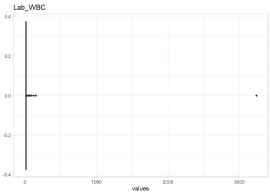
```
## Warning: Removed 443 rows containing non-finite values (stat_boxplot).
```



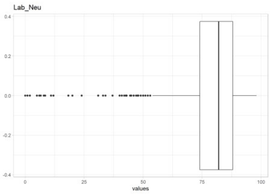
```
## Warning: Removed 466 rows containing non-finite values (stat_boxplot).
```



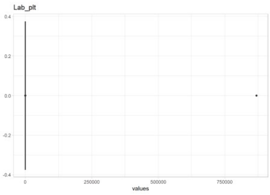
```
## Warning: Removed 424 rows containing non-finite values (stat_boxplot).
```



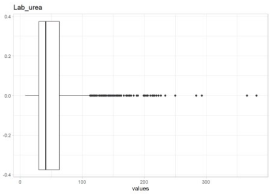
```
## Warning: Removed 557 rows containing non-finite values (stat_boxplot).
```



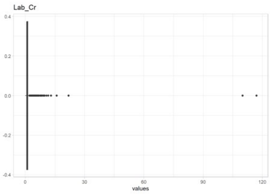
```
## Warning: Removed 808 rows containing non-finite values (stat_boxplot).
```



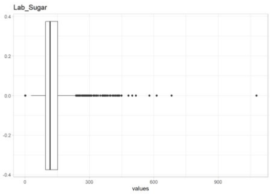
```
## Warning: Removed 660 rows containing non-finite values (stat_boxplot).
```



```
## Warning: Removed 744 rows containing non-finite values (stat_boxplot).
```



```
## Warning: Removed 813 rows containing non-finite values (stat_boxplot).
```



High `Lab_Hb` levels

In conventional units, a healthy female has 12-16 g/DL of haemoglobin. In SI units, a healthy female has 120-160 g/L of haemoglobin. Based on the distribution of values `Lab_hb`, the outliers are likely measured in g/L (e.g. 111-149) will need to be converted to g/DL (e.g. 11.1-14.9).

```
(df %>% filter(Lab_Hb>100) %>% select(Lab_Hb))
```

```
## # A tibble: 6 x 1
##   Lab_Hb
##
## 1    114
## 2    111
## 3    132
## 4    112
## 5    135
## 6    149
```

```
# insert decimal point
df< -df %>% mutate(Lab_Hb= if_else(Lab_Hb>100, Lab_Hb/10, Lab_Hb))
```

Low `Lab_Neu`

Neutrophil <40 are considered below normal limits. The initially hypothesis is that these patients with low neutrophil have either HIV or on immunosuppression drugs. However, majority of these patients have neither. Although, the initial hypothesis was incorrect, there are other differentials for low neutrophil levels. Considering, the number of these outliers is small (n=2

```
df %>% select (Lab_Neu, Hx_HIV, Hx_immune) %>% filter(Lab_Neu<35)
```

```
## # A tibble: 22 x 3
##    Lab_Neu Hx_HIV      Hx_immune
##
##  1      11 No          No
##  2      12 No          Yes
##  3       0 Unavailable No
##  4      24 No          No
##  5       6 No          No
##  6      34 No          No
##  7       0 No
##  8       2 No          Yes
##  9       5 Unavailable No
## 10       1 No          No
## # ... with 12 more rows
```

High `Lab_Sugar`

It is plausible to have very high sugar levels if the patient has diabetes. All of these outliers have diabetes thus the outlier values are plausible.

```
df %>% filter(Lab_Sugar>450) %>% count(Hx_diabetes)
```

```
## # A tibble: 1 x 2
##   Hx_diabetes     n
##
## 1 Yes             7
```

10 `CS_` cultures related category

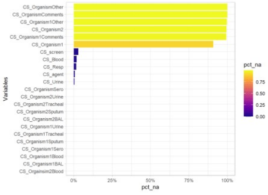
There are 23 variables under `CS` and the most important variables are `CS_Organism1` and `CS_Organism2` as they indicate which organism is causing the CAP. The majority of the other `CS` variables are methods to identify the organism. However, there are >90% missing values for `CS_Organism1` and `CS_Organism2` thus the methods of identifying the organisms th

```
(dtype(df, "CS"))
```

```
## tibble [2,112 x 23] (S3: tbl_df/tbl/data.frame)
##  $ CS_Resp             : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_Blood            : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_Urine            : chr [1:2112] "Yes" "Yes" "Yes" "Yes" ...
##  $ CS_screen           : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_agent            : chr [1:2112] "No" "No" "No" "Yes" ...
##  $ CS_Organism1        : chr [1:2112] NA NA NA "Streptococcus pneumoniae" ...
##  $ CS_Organism1Blood   : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_Organism1Sputum  : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_Organism1Tracheal: chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_Organism1BAL     : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_Organism1Urine   : chr [1:2112] "No" "No" "No" "Yes" ...
##  $ CS_Organism1Sero    : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_Organism1Other   : chr [1:2112] NA NA NA NA ...
##  $ CS_Organism1Comments: chr [1:2112] NA NA NA NA ...
##  $ CS_Organism2        : chr [1:2112] NA NA NA NA ...
##  $ CS_Orgainsim2Blood  : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_Organism2Sputum  : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_Organism2Tracheal: chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_Organism2BAL     : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_Organism2Urine   : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_OrganismSero     : chr [1:2112] "No" "No" "No" "No" ...
##  $ CS_OrganismOther    : chr [1:2112] NA NA NA NA ...
##  $ CS_OrganismComments : chr [1:2112] NA NA NA NA ...
```

```
## NULL
```

```
(eda_n_NAplt(df, "CS"))
```



```
df< -select(df, -starts_with("CS"))
```

11 `Abx_` antibiotics related category

```
dtype(df, "Abx")
```

```
## tibble [2,112 x 35] (S3: tbl_df/tbl/data.frame)
##  $ Abx_AmoxicillinSulbactam           : chr [1:2112] "No" "Yes" "Unavailable" "No" ...
##  $ Abx_AmoxicillinSulbactamOral       : chr [1:2112] "No" "No" "Unavailable" "No" ...
##  $ Abx_AmoxicillinSulbactamNonoral    : chr [1:2112] "No" "No" "Unavailable" "No" ...
##  $ Abx_AmoxicillinSulbactamNonoralStart: chr [1:2112] NA NA NA NA ...
##  $ Abx_AmoxicillinSulbactamNonoralEnd  : chr [1:2112] NA NA NA NA ...
##  $ Abx_Ampicillin                     : chr [1:2112] "No" "No" "Unavailable" "No" ...
##  $ Abx_AmpicillinStart                : chr [1:2112] NA NA NA NA ...
##  $ Abx_AmpicillinEnd                  : chr [1:2112] NA NA NA NA ...
##  $ Abx_AmpicillinSulbactam            : chr [1:2112] "No" "No" "Unavailable" "No" ...
##  $ Abx_Azithromycin                   : chr [1:2112] "No" "Yes" "Yes" "Yes" ...
##  $ Abx_Ceftriaxone                    : chr [1:2112] "Yes" "No" "Yes" "No" ...
##  $ Abx_Cefotaxime                     : chr [1:2112] "No" "No" "Yes" "Yes" ...
##  $ Abx_ClarithromycinOral             : chr [1:2112] "No" "No" "Unavailable" "No" ...
##  $ Abx_Cefepime                       : chr [1:2112] "No" "No" "Unavailable" "No" ...
##  $ Abx_CefepimeStart                  : logi [1:2112] NA NA NA NA NA NA ...
##  $ Abx_CefepimeEnd                    : logi [1:2112] NA NA NA NA NA NA ...
##  $ Abx_ClarithromycinIV               : chr [1:2112] "No" "No" "Unavailable" "No" ...
##  $ Abx_ClarithromycinIVStart          : chr [1:2112] NA NA NA NA ...
##  $ Abx_ClarithromycinIVEnd            : chr [1:2112] NA NA NA NA ...
##  $ Abx_Doxycycline                    : chr [1:2112] "No" "No" "Unavailable" "No" ...
##  $ Abx_DoxycyclineStart               : chr [1:2112] NA NA NA NA ...
##  $ Abx_DoxycyclineEnd                 : chr [1:2112] NA NA NA NA ...
##  $ Abx_Levofloxacin                   : chr [1:2112] "No" "No" "Yes" "Yes" ...
##  $ Abx_Moxifloxacin                   : chr [1:2112] "Yes" "No" "Unavailable" "No" ...
##  $ Abx_Piperacillin                   : chr [1:2112] "No" "No" "Unavailable" "No" ...
##  $ Abx_PiperacillinStart              : chr [1:2112] NA NA NA NA ...
##  $ Abx_PiperacillinEnd                : chr [1:2112] NA NA NA NA ...
##  $ Abx_Trimethoprim                   : chr [1:2112] "No" "No" "Unavailable" "No" ...
##  $ Abx_TrimethoprimStart              : chr [1:2112] NA NA NA NA ...
##  $ Abx_TrimethoprimEnd                : chr [1:2112] NA NA NA NA ...
##  $ Abx_OtherYN                        : chr [1:2112] "No" "No" "No" "No" ...
##  $ Abx_OtherDetail                    : chr [1:2112] NA NA NA NA ...
##  $ Abx_Class                          : chr [1:2112] "Beta-lactams + Quinolones" "Beta-
lactams + Macrolides" "Beta-lactams + Quinolones" "Macrolides" ...
##  $ Abx_ClassOther                     : chr [1:2112] NA NA NA NA ...
##  $ Abx_Duration                       : num [1:2112] 14 10 5 16 8 13 10 10 10 10 ...
```

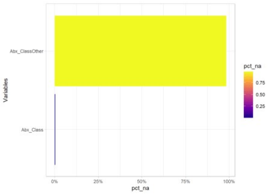
The antibiotics category can be divided into 3 sub-categories:

i. Class of empirical antibiotics given
ii. Antibiotics given
iii. Duration of antibiotics

11i Class of empirical antibiotics given

Majority of the `Abx_ClassOther` are `NA` because they have values in `Abx_Class`.

```
eda_n_NAplt(df, "Abx_Class")
```



Majority of the `NA` `Abx_ClassOther` are `NA` because they have values in `Abx_Class`.

```
df %>% select(starts_with("Abx_Class")) %>% filter(is.na(Abx_ClassOther)) %>% count(Abx_Class)
```

```
## # A tibble: 7 x 2
##   Abx_Class                     n
##
```

```
## 1 Beta-lactams               1108
## 2 Beta-lactams + Macrolides   736
## 3 Beta-lactams + Quinolones    59
## 4 Macrolides                   79
## 5 Other                         2
## 6 Quinolones                   89
## 7                            6
```

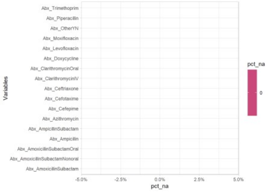
Map values from `Abx_Class` to replace `NA` values in `Abx_ClassOther`. After using `Abx_Class` to expand `Abx_ClassOther`, `Abx_Class` will be dropped. Rename the updated `Abx_ClassOther` to `Abx_ClassUpdated` for more intuitive understanding of variable name

```
df< -df %>% mutate(Abx_ClassOther= case_when(
  Abx_Class=="Beta-lactams" & is.na(Abx_ClassOther) ~ "Beta-lactams",
  Abx_Class=="Beta-lactams + Macrolides" & is.na(Abx_ClassOther) ~"Beta-lactams + Macrolides",
  Abx_Class=="Beta-lactams + Quinolones" & is.na(Abx_ClassOther) ~ "Beta-lactams + Quinolones",
  Abx_Class=="Macrolides" & is.na(Abx_ClassOther) ~ "Macrolides",
  Abx_Class== "Other" & is.na(Abx_ClassOther) ~ "Other",
  Abx_Class== "Quinolones" & is.na(Abx_ClassOther) ~ "Quinolones",
  T ~ Abx_ClassOther
)) %>%
  # remove Abx_Classother
  select(-Abx_Class) %>%
  # rename
  rename(Abx_ClassUpdated= Abx_ClassOther)
```

11ii Antibiotics given

There are no missing values for antibiotics given

```
df %>% select(starts_with("Abx")) %>% select(- c(ends_with("Start") | ends_with("End") |
Abx_Duration | starts_with("Abx_Class")|ends_with("Detail")))  %>% eda_n_NAplt("Abx")
```



Number of antibiotics given

From the antibiotics given, the number of antibiotics given can be calculated. There are 4 observations with `NA` values being calculated. These observations shall be examined to see if there are missing values or if no antibiotics were given to begin with. (Perhaps, the doctor had high index of suspicion it was a viral CAP. In such situation, antibiotics would be ineffective

```
## function to extract abx taken by pt (lond df)
abx_taken_Longdf< - function(dfr){
  # select case number and abx col
  dfr %>% select(Pt_CaseNumber, starts_with("Abx")) %>%
  # remove unrelated abx columns
  select(- c(ends_with("Start") | ends_with("End") | Abx_Duration |
starts_with("Abx_Class")|ends_with("Detail"))) %>%
  # into long df
    pivot_longer(-Pt_CaseNumber, names_to="Abx_type", values_to="Used") %>%
  # filter abx taken
    filter(Used=="Yes")
}

## join no of abx taken w main df
df< -left_join(x= df,
  y=df %>% abx_taken_Longdf() %>%  group_by(Pt_CaseNumber) %>% count(Used, name= "New_Abx_no")
%>% ungroup(),
        by= "Pt_CaseNumber")

df %>% count(New_Abx_no)
```

```
## # A tibble: 8 x 2
##   New_Abx_no     n
##
## 1          1   778
## 2          2   854
## 3          3   347
## 4          4   121
## 5          5     6
## 6          6     1
## 7          7     1
## 8         NA     4
```

```
# convert abx as integer to numeric as Error: Problem with `mutate()` input `Abx_no`. x must be
a double vector, not an integer vector. i Input `Abx_no` is `case_when(...)`.
df< - df %>% mutate(New_Abx_no= as.numeric(New_Abx_no))
```

Patient `254`, `916`, `964` did not receive any antibiotic. The number of antibiotics taken will be `0` and the antibiotics duration will also be `0`. Patient `1864` received `Macrolides` class antibiotics as an empirical treatment. Fill the number of antibiotics taken as `1` and fill up other antibiotics taken `Abx_OtherYN` as `Yes` and fill up details of other antibiotics taken `Abx_OtherD`

```
df%>% filter(is.na(New_Abx_no)) %>% select(Pt_CaseNumber, starts_with("Abx")) %>% select(-
c(ends_with("Start") | ends_with("End") )) %>%  DT::datatable(rownames = F, options =
list(searchHighlight = TRUE, paging= T))
```

\n \n

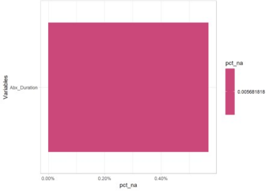
Pt_CaseNumber< \/th>\n	Abx_AmoxicillinSulbactam< \/th>\n	Abx_AmoxicillinSulbactamOral< \/th>\n	Abx_ AmoxicillinSulbactamNonoral< \/th>\n	Abx_Ampicillin< \/th>\n	Abx_AmpicillinSulbactam< \/th>\n	Abx_Azithromycin< \/th>\n	Abx_Ceftriaxone< \/th>\n	Abx_Cefotaxime< \/th>\n	Abx_ClarithromycinOral< \/th>\n	Abx_Cefepime< \/th>\n	Abx_ClarithromycinIV< \/th>\n	Abx_Doxycycline< \/th>\n	Abx_Levofloxacin< \/th>\n	Abx_Moxifloxacin< \/th>\n	Abx

\n

11iii Duration of antibiotics

12 observations had `NA` antibiotic duration. Mostly like due to some data calculation or data entry error as only 3 observations in the entire dataset did not receive antibiotics.

```
(eda_n_NAplt(df, "Abx_Duration"))
```



```
(df %>% filter(is.na(Abx_Duration)) %>% distinct(Pt_CaseNumber) %>% count())
```

```
## # A tibble: 1 x 1
##       n
##
## 1    12
```

All of the patients with missing antibiotic duration took antibiotics.

```
df %>% filter(is.na(Abx_Duration)) %>% select(Pt_CaseNumber, New_Abx_no)
```

```
## # A tibble: 12 x 2
##    Pt_CaseNumber New_Abx_no
##
##  1           289          2
##  2           378          1
##  3           453          3
##  4           703          1
##  5           720          1
##  6           832          1
##  7           855          2
##  8          1198          4
##  9          1393          4
## 10          1523          1
## 11          1713          3
## 12          2024          2
```

An attempt is made to calculate the duration using start and end dates of the antibiotics given.
3/12 patients with missing antibiotics duration had the start dates of their antibiotics captured. However, these patients took other antibiotics which did not have the start dates captured. We are unable to impute any of the missing antibiotic duration by calculating the difference in antibiotic start and end dates. We will impute the missing antibiotic duration by other means

```
#  abx with start  dates
abx_date< -df %>% select(ends_with("Start")) %>% colnames() %>% str_replace("Start","")

# type of abx taken for pt w m/s abx duration
abx_ms< -
  # filter pt with m/s abx_duration
  df %>% filter(is.na(Abx_Duration)) %>%
  # used above function to find out abx taken
  abx_taken_Longdf() %>%
  # distinct abx taken by this group of pts
  distinct(Abx_type) %>% pull()

# types of abx taken for pt w m/s abx duration which have date of abx captured
abx_msAndDateStarted< -intersect(abx_date, abx_ms)

# pt w m/s abx duration who took abx with at least one abx start date
(df %>% filter(is.na(Abx_Duration)) %>%
  abx_taken_Longdf() %>%
  group_by(Pt_CaseNumber) %>%  filter(any(Abx_type==abx_msAndDateStarted[[1]]) |
any(Abx_type==abx_msAndDateStarted[[2]])) %>%   summarise(n=n(), .groups="drop"))
```

```
## # A tibble: 3 x 2
##   Pt_CaseNumber     n
##
## 1           453     3
## 2          1198     4
## 3          1393     4
```

```
# remove start and end date
df< -df %>% select(- c(ends_with("Start") | ends_with("End")))
```

12 `Care_` continuum of care status category

```
(dtype(df, "Care"))
```

```
## tibble [2,112 x 7] (S3: tbl_df/tbl/data.frame)
##  $ Care_admit           : chr [1:2112] "Yes" "No" "Yes" "Yes" ...
##  $ Care_ICU             : chr [1:2112] "No" "No" "No" "No" ...
##  $ Care_breathingAid    : chr [1:2112] "No" "No" "No" "No" ...
##  $ Care_breathingAidType: chr [1:2112] NA NA NA NA ...
##  $ Care_BPSupport       : chr [1:2112] "No" "No" "Yes" "No" ...
##  $ Care_daysUnfit       : num [1:2112] 10 10 10 15 NA 12 15 6 12 NA ...
##  $ Care_GP/OutptVisit   : num [1:2112] 2 4 1 1 1 2 2 1 3 2 ...
```

```
## NULL
```

```
(eda_c(df, "Care"))
```

```
## $Care_admit
## .x
##          No Unavailable         Yes
##         631           2        1479           0
##
## $Care_ICU
## .x
##          No Unavailable         Yes
##        1729          53         330           0
##
## $Care_breathingAid
## .x
##          No Unavailable         Yes
##        1928          60         124           0
##
## $Care_breathingAidType
## .x
##          ARM CPAP/Bilevel        Other
##           91           26            5         1990
##
## $Care_BPSupport
## .x
##          No Unavailable         Yes
##        1930          85          97           0
##
## $Care_daysUnfit
## .x
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
##    9   17   30   45   47  116   37  298   62   25  408   22   76   23  210  186
##   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31
##   18   11   17    6  141   58    9    4    7   25    2    2    1    1   67    2
##   33   35   36   37   40   45   47   48   50   53   58   59   60   65   69   75
##    4    6    1    1    8    5    1    1    3    1    1    1    3    1    1    1
##   80   99  181
##    1   20    1   69
##
## $`Care_GP/OutptVisit`
## .x
##    0    1    2    3    4    5    6    7   10   15   21   99
##   57 1460  307  173   56   14    2    5    1    1    1   14   21
```

replace 99

Again `99` appears as outliers for `Care_daysUnfit` and `Care_GP/OutptVisit`. `99` will be replaced with `NA`.

```
df< -df %>% mutate(Care_daysUnfit= na_if(Care_daysUnfit, 99),
              `Care_GP/OutptVisit`= na_if(`Care_GP/OutptVisit`, 99))
```

Admission status

`Care_admit` indicates if the patient was admitted to a hospital and `Care_ICU` indicates if patient had an ICU stay. 324 patients who were hospitalized also had ICU stay. The labels in `Care_admit` will include details to reflect patient who were admitted AND had ICU stay (label as `Yes (w ICU)`). After using information from `Care_ICU` to expand `Care_admit`, `Care

```
(table(df$Care_admit, df$Care_ICU, useNA = "always"))
```

```
##
##                  No Unavailable  Yes
##   No            584          47    0    0
##   Unavailable     0           2    0    0
##   Yes          1145           4  330    0
##                 0           0    0    0
```

```
df< -df %>% mutate(Care_admit= case_when(
  Care_admit=="Yes" & Care_ICU=="Yes" ~ "Yes (w ICU)",
  T~ Care_admit)) %>%
  select(-Care_ICU)

(df %>% count(Care_admit, name = "new_tally"))
```

```
## # A tibble: 4 x 2
##   Care_admit  new_tally
##
## 1 No                631
## 2 Unavailable         2
## 3 Yes              1149
## 4 Yes (w ICU)       330
```

Breathing aid

`Care_breathingAid` indicates if patient in ICU used any breathing aids. `Care_ breathingAidType` details the type of breathing aids used.
Details from `Care_breathingAidType` will be integrated into `Care_breathingAid` and the `Care_breathingAidType` will be dropped.

```
(table(df$Care_breathingAid, df$Care_breathingAidType, useNA = "always"))
```

```
##
##                 ARM CPAP/Bilevel Other
##   No              0            0     0 1928
##   Unavailable     0            0     0   60
##   Yes            91           26     5    2
##                 0            0     0    0
```

```
df< -df %>% mutate(Care_breathingAid= case_when(
  Care_breathingAid=="Yes" & Care_breathingAidType=="ARM" ~ "ARM",
  Care_breathingAid=="Yes" & Care_breathingAidType=="CPAP/Bilevel" ~ "CPAP/Bilevel",
  Care_breathingAid=="Yes" & Care_breathingAidType=="Other" ~ "Other",
  Care_breathingAid=="Yes" & is.na(Care_breathingAidType) ~ "Other",
  T~ Care_breathingAid)) %>% select(-Care_breathingAidType)

(count(df, Care_breathingAid, name = "new_tally"))
```

```
## # A tibble: 5 x 2
##   Care_breathingAid new_tally
##
## 1 ARM                      91
## 2 CPAP/Bilevel             26
## 3 No                     1928
## 4 Other                     7
## 5 Unavailable              60
```

13 `V_` vaccine related category

```
(dtype(df, "V"))
```

```
## tibble [2,112 x 2] (S3: tbl_df/tbl/data.frame)
##  $ V_pneumococcal: chr [1:2112] "Yes" "Yes" "No" "No" ...
##  $ V_flu         : chr [1:2112] "Yes" "Yes" "No" "Yes" ...
```

```
## NULL
```

```
(eda_c(df, "V"))
```

```
## $V_pneumococcal
## .x
##          No Unavailable         Yes
##        1721          26         365           0
##
## $V_flu
## .x
##          No Unavailable         Yes
##        1448          27         637           0
```

Currently each `V_` column indicates if the patient has received that particular vaccine. As there are only two columns, the values of both columns will be united to indicate which vaccines the patient has received.

```
df< -df  %>%
  mutate(V_pneumococcal= if_else(V_pneumococcal=="Yes", "pneumococcal,", ""),
              V_flu=if_else(V_flu=="Yes", "flu", "")) %>%
  unite(V_vaccine, V_pneumococcal, V_flu, sep = "", remove = T) %>%
  mutate(V_vaccine= if_else(V_vaccine=="", "no/unavailable", V_vaccine))

(count(df, V_vaccine))
```

```
## # A tibble: 4 x 2
##   V_vaccine            n
##
## 1 flu                313
## 2 no/unavailable    1434
## 3 pneumococcal,       41
## 4 pneumococcal,flu   324
```

Wrap up

The original dataset had 2302 rows and 176 columns, after EDA the dataset has 2112 rows and 78 columns. More than half of the columns were removed and compressed via EDA.

```
# Clean up intermediate columns created during EDA
df< -df %>% select(-Used) %>% rename(Abx_no=New_Abx_no)

dim(df)
```

```
## [1] 2112   78
```

The cleaned up dataset is ready for some action. In the next post, some feature engineering will be done.

```
df  %>%  DT::datatable(rownames = F, options = list(searchHighlight = TRUE, paging= T))
```