**Introduction**

The aim of this series of blogs is to do time series forecasting with libraries that conform to tidyverse principles and there are two of these time series meta-packages

1. modeltime which is created to be the time series equivalent of tidymodels
2. fpp3 which is created to do tidy time series and has been nicknamed the tidyverts. Modelling can be done with fpp3 but it is limited to sequential models and mostly uses classical forecasting approaches.

The [dataset is monthly admission to Singapore public acute adult hospitals](https://www.tablebuilder.singstat.gov.sg/publicfacing/mainMenu.action). Singapore’s hospitals are grouped into three healthcare clusters based on geographical region.   
The hospital network in the west is known as National University Hospital System (NUHS) and it has three acute hospitals, Alexandra Hospital (AH); National University Hospital (NUH); Ng Teng Fong General Hospital (NTFGH).   
The hospital group in the north is known as National Healthcare Group (NHG) and it has two acute hospitals, Tan Tock Seng Hospital (TTSH); Khoo Teck Puat Hospital (KTPH).   
The healthcare cluster in the south and east is known as SingHealth System (SHS) and it has it has three acute hospitals, Changi General Hospital (CGH); Sengkang Hospital (SKH); Singapore General Hospital (SGH).   
The dataset starts from Jan 2016 and ends in Feb 2021. 2016 was selected as the starting point as the number of hospitals in 2015 varied (NGTFH opened in end Jun 15 and AH closed from end Jun to Aug 15). Moreover, the time series would have more than 4 years of observations, which should be adequate for time series forecasting. Feb 21 was selected as the end point as that was the latest data available.

library(tidyverse)

raw<- read\_csv("<https://raw.githubusercontent.com/notast/hierarchical-forecasting/main/stat_sg.csv>", na= "na", skip=4, n\_max=9) %>%

select(-X64) %>%

# remove \*

mutate(Variables= str\_remove(Variables, " \\\*")) %>%

# remove Public Sector Hospital Admissions, the value is inaccurate, likely it included children's and mental hospitals

filter(Variables != "Public Sector Hospital Admissions") %>%

# longer

pivot\_longer(-Variables, names\_to="Date", values\_to="Admission") %>%

# convert date format

mutate(Date=parse\_date(Date, "%Y %b")) %>%

# short hand

mutate(Variables= recode(Variables, # recode old=new

`Alexandra Hospital` = "AH",

`Changi General Hospital` = "CGH", `Khoo Teck Puat Hospital` = "KTPH",

`National University Hospital` = "NUH", `Ng Teng Fong General Hospital` = "NTFGH",

`Sengkang General Hospital` = "SKH", `Singapore General Hospital` = "SGH",

`Tan Tock Seng Hospital`= "TTSH")) %>%

# cluster

mutate(Cluster= case\_when(

Variables %in% c("CGH", "SKH", "SGH") ~ "SHS",

Variables %in% c("NUH", "AH", "NTFGH") ~ "NUHS",

Variables %in% c("TTSH", "KTPH") ~ "NHG")) %>%

# rename

rename(Hospital= Variables)

# recalculate total hospital admissions

df<-raw %>% group\_by(Date) %>% summarise(Admission=sum(Admission, na.rm = T)) %>% mutate(Cluster= "National", Hospital= "National") %>% bind\_rows(raw)

df\_cluster<- df %>% group\_by(Cluster, Date) %>% summarise(Admission= sum(Admission, na.rm = T), .groups= "drop")

**Visualization**

The agenda for this post is EDA before forecasting is attempted.

library(fpp3)

library(timetk)

df\_tsib<- df %>%

mutate(Date= yearmonth(as.character(Date))) %>%

# if there are two index, need to use group\_by and summarise to isolate a specific index

as\_tsibble(key= c(Hospital), index= Date)

There are at least 5 concepts that we should visualize before forecasting a time series:

1. Trend
2. Seasonality
3. Anomaly
4. Lags (which is the foundation for autocorrelation and partial autocorrelation)
5. Correlation of time series features

Hospital level data was visualized with all the above concepts while cluster level data was visualized when it was appropriate. Abbreviations of the hospital and cluster names were used to keep the plots neat. The grand total admission is labelled as National at hospital level and is labelled as National at cluster level.

**1. Trend**

Trend refers to the long-term increase or decrease in the data, it indicates any changing direction of the data

df %>% group\_by(Hospital) %>% plot\_time\_series( Date, Admission, .interactive = F, .legend\_show = F, .facet\_ncol = 3, .title="Time Series (hospital level)")

df\_cluster %>% group\_by(Cluster) %>%

plot\_time\_series( Date, Admission, .interactive = F, .legend\_show = F, .facet\_ncol = 2, .title= "Time Series (cluster level)")

**2. Seasonality**

Seasonality allow us to view patterns over time and identify time periods in which the patterns change.

timetk’s plot\_seasonal\_diagnostics is a convenient function to plot the observations by month, year and quarter. Unfortunately, it has limited flexibility to adjust the aesthetics and the plots can be overcrowded.

df%>% group\_by(Hospital) %>% plot\_seasonal\_diagnostics(Date, Admission, .interactive=F, .feature\_set="month.lbl")

## Warning: Removed 30 rows containing non-finite values (stat\_boxplot).

feasts’s has gg\_season to plot seasonal graphs.

* There are fewer admissions in Feb.

df\_tsib %>%

gg\_season(Admission, period="year")

However, I could not change the temporal granularity.

df\_tsib %>% gg\_season(Admission, period="month")

## Error: The data must contain at least one observation per seasonal period.

Additionally, I wanted outliers to be highlighted which would have been easier with box plots like in timetk::plot\_seasonal\_diagnostics. I created plots similar to timetk::plot\_seasonal\_diagnostics but allowed faceting for better aesthetics.

* Again, fewer admissions in Feb is seen, likely for two reasons. Firstly, Feb has the shortest month and Chinese Lunar New Year tends to happen during Feb.
* There are more admissions in the final quarter of the year, mostly from Oct and Dec.

fun\_season<- function(dat, var, f, t){

ggplot(dat, aes(!!{{var}}, Admission))+ geom\_boxplot() + facet\_wrap(vars({{f}}), scales = "free\_y") +scale\_x\_discrete(guide = guide\_axis(angle = 45)) + labs(title= paste(var, t))}

c("Year", "Quarter", "Month") %>% syms() %>%

map(., ~fun\_season(

dat= df %>% mutate(Year= as.factor(year(Date)),

Quarter= as.factor(quarter(Date)),

Month= as.factor(month(Date))),

.x, Hospital, "Seasonality (hospital level)"))

## [[1]]

##

## [[2]]

##

## [[3]]

c("Y", "Q", "M") %>% syms() %>%

map(., ~fun\_season(

dat= df\_cluster %>% mutate(Y= as.factor(year(Date)),

Q= as.factor(quarter(Date)),

M= as.factor(month(Date))),.x, Cluster, "Seasonality (cluster level)"))

## [[1]]

##

## [[2]]

##

## [[3]]

A useful summary plot covering various temporal granularity was achieved with gg\_subseries.

* The low admission during Feb is more obvious in this plot as the horizontal blue line highlighted the mean.

df\_tsib %>%gg\_subseries(y=Admission)

**Trend and seasonality**

A more objective manner to determine trend and seasonality to use to values from STL, more specifically trend\_strength and seasonal\_strength\_year. 0 implies weak strength while 1 implies 1 strong strength. Afpp3 package, fabletools, calculates STL values from the feasts package and outputs it into a dataframe. I piped the output into a plot.

* Hospitals have less than moderate seasonality, with CGH having the highest seasonal strength of 0.66.
* CGH has the lowest trend strength which is also observed in the trend plot at the start of the post where CGH has a flat loess line. SKH has the highest trend strength which is also observed in the very first plot where SKH had a steep loess line.

df\_tsib %>% features(Admission, feature\_set(tags = "stl")) %>% ggplot(aes(x=trend\_strength, y=seasonal\_strength\_year, col=Hospital, label=Hospital)) +

geom\_point() +

geom\_text( check\_overlap=T, hjust=1, vjust=1)+

scale\_x\_continuous(limits = c(0, 1)) + scale\_y\_continuous(limits = c(0, 1))

**3. Anomaly**

timetk has a useful function to flag up anomaly observations, plot\_anomaly\_diagnostics.

df %>% mutate(Admission=replace\_na(Admission, 0)) %>% # need to replace NA or calculation error

group\_by(Hospital) %>%

plot\_anomaly\_diagnostics(.date\_var = Date, .value= Admission, .facet\_ncol=3, .facet\_scales="free\_y", .ribbon\_alpha=.25, .message=F, .legend\_show=F, .title= "Anomaly (hospital level)", .interactive=F)

df\_cluster %>% group\_by(Cluster) %>%

plot\_anomaly\_diagnostics(.date\_var = Date, .value= Admission, .facet\_ncol=3, .facet\_scales="free\_y", .ribbon\_alpha=.25, .message=F, .legend\_show=F, .title= "Anomaly (cluster level)", .interactive=F)

Most of the anomaly detected occurred during the peak of COVID19 pandemic from Jan 20- Jul 20. The anomaly in 2018 came from SKH and was not observed in other hospitals nor at a more aggregated cluster level. The anomaly was likely due to the change in the number of admissions before and after the hospital was opened in Jul 18.

df %>% mutate(Admission=replace\_na(Admission, 0)) %>% group\_by(Hospital) %>% tk\_anomaly\_diagnostics(.date\_var = Date, .value=Admission) %>% ungroup() %>% filter(anomaly== "Yes") %>% count(Date, sort=T)

## # A tibble: 16 x 2

## Date n

## <date> <int>

## 1 2020-04-01 5

## 2 2020-03-01 4

## 3 2020-05-01 3

## 4 2020-01-01 2

## 5 2020-02-01 2

## 6 2018-04-01 1

## 7 2018-05-01 1

## 8 2018-06-01 1

## 9 2018-07-01 1

## 10 2018-08-01 1

## 11 2018-10-01 1

## 12 2018-11-01 1

## 13 2018-12-01 1

## 14 2019-01-01 1

## 15 2020-06-01 1

## 16 2020-07-01 1

**Conclusion**

EDA revealed that there was an upward trend of admissions till the first half of 2020 which was flagged up as anomaly observations. A dummy variable for this heightened COVID19 period would be considered when forecasting. Seasonality varied though a dipped in admissions during Feb was common, this observation should be included when forecasting.