SNCF Trains analysis

Code

Hide

```
options(warn=-1)
library(readxl)
#df_raw = read.csv("full_trains.csv")
df_raw = read_xlsx(path="regularite-mensuelle-tgv-aqst_upd.xlsx")
library(igraph)
library(ggraph)
library(tidyverse)
library(scales)
library(patchwork)
library(dplyr)
library(viridis)
library(hrbrthemes)
```

Select everything except comments for cancellation, departure and arrival delays

Hide

```
df = df_raw %>%
  select(-c(comment_cancellations,comment_delays_at_departure,comment_delays_on_arrival))

df = df %>%
  mutate(pct_late_departure = num_late_at_departure/total_num_trips,
         mean_monthly_pct_cancelled = num_of_cancelled_trains/total_num_trips)
```

Check for missing values

Hide

```
na_count = data.frame(na_sum = colSums(is.na(df)))%>%
arrange(desc(na_sum))

na_count %>%
filter(na_sum !=0)
```

	na_sum <dbl>
pct_late_departure	73
mean_monthly_pct_cancelled	10

2 rows

Hide

```
head(df)
```

year <dbl>	mo… <dbl>	service <chr>	departure_station <chr>	arrival_station <chr>	journey_time_avg <dbl>	total_num_trips <dbl>
2018	1	National	BORDEAUX ST JEAN	PARIS MONTPARNASSE	141	870
2018	1	National	LA ROCHELLE VILLE	PARIS MONTPARNASSE	165	222
2018	1	National	PARIS MONTPARNASSE	QUIMPER	220	248
2018	1	National	PARIS MONTPARNASSE	ST MALO	156	102
2018	1	National	PARIS MONTPARNASSE	ST PIERRE DES CORPS	61	391
2018	1	National	QUIMPER	PARIS MONTPARNASSE	223	256

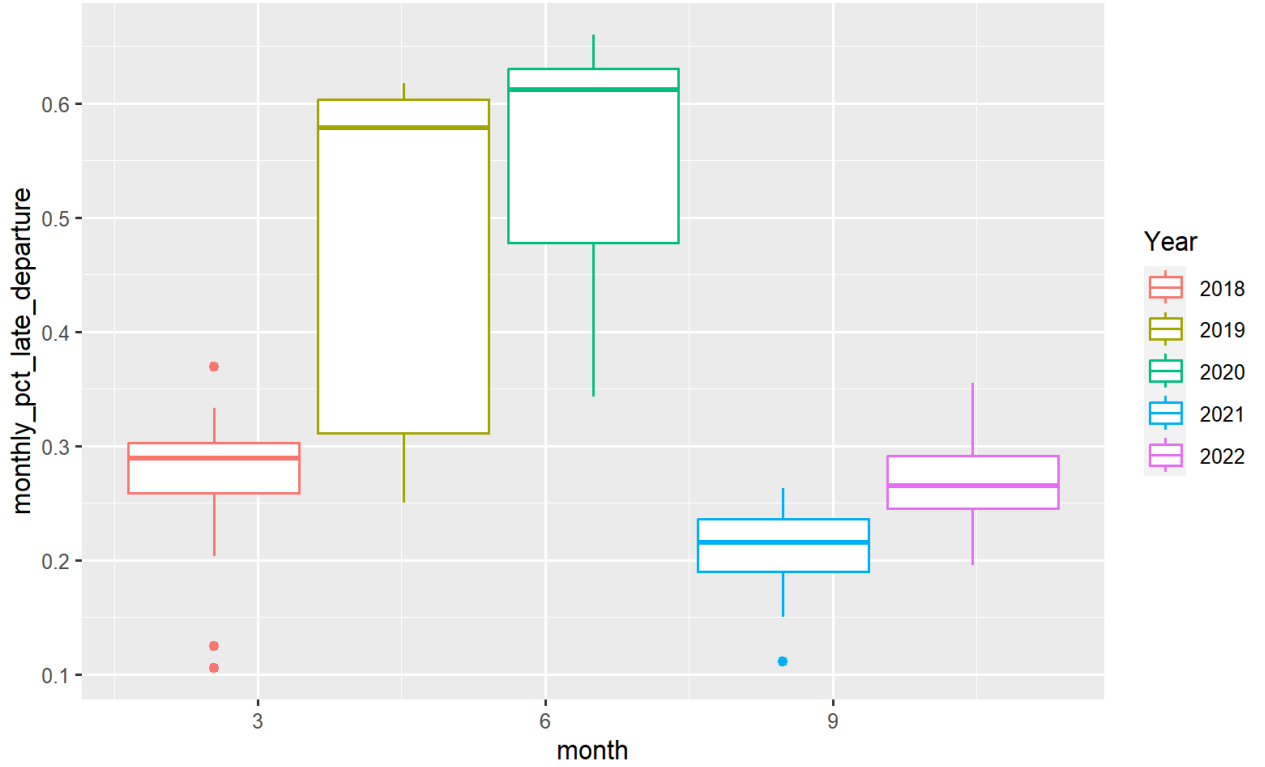
6 rows | 1-7 of 26 columns

Now, let's look a boxplot of % late departures per year

Hide

```
df %>%
  mutate(pct_late_departure = num_late_at_departure/total_num_trips) %>%
  group_by(year,month) %>%
  summarise(monthly_pct_late_departure = mean(pct_late_departure, na.rm = TRUE)) %>%
  ggplot(aes(month,monthly_pct_late_departure,color = factor(year)))+
    geom_boxplot(size = 0.5)+
    labs(color = "Year")
```

```
`summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

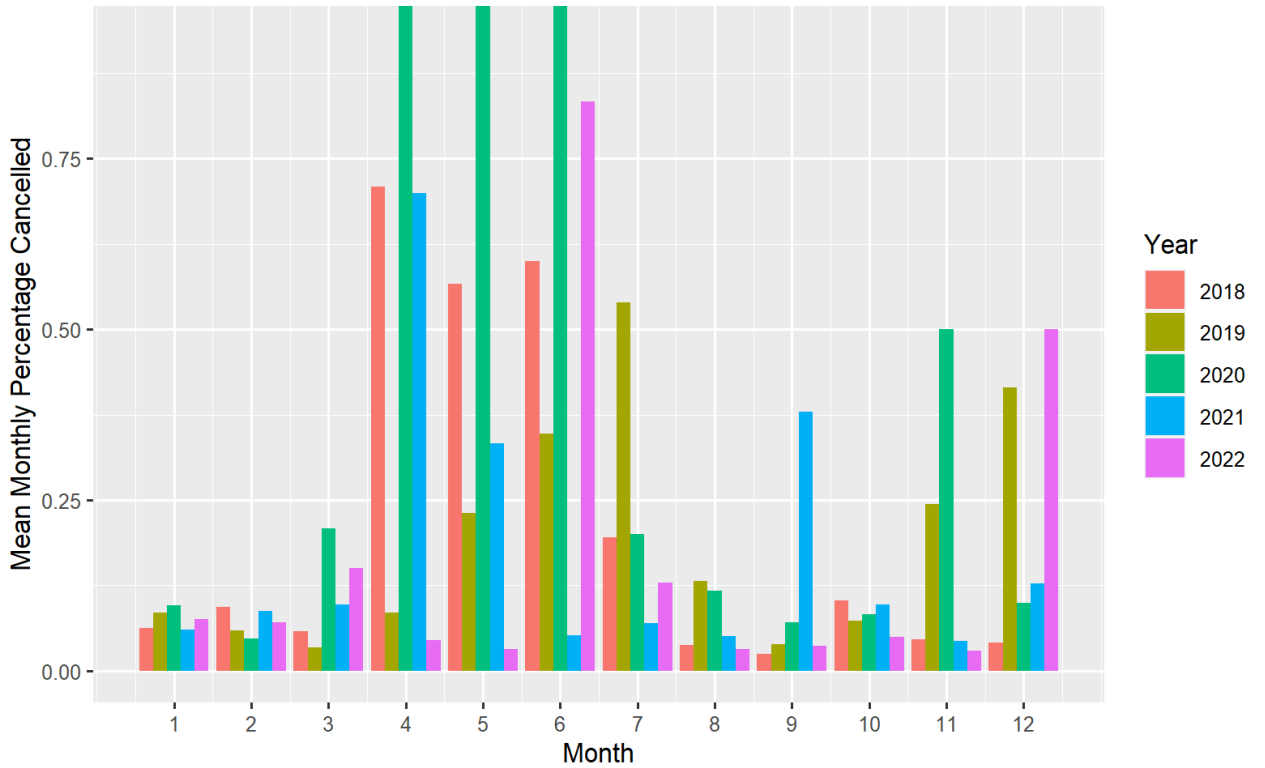


2019 and 2020 medians are around 60%.

Create a bar chart by month for all years.

Hide

```
df %>%
  ggplot(aes(x = month, y = mean_monthly_pct_cancelled, fill = factor(year))) +
  geom_col(position = "dodge", na.rm = TRUE) +
  labs(x = "Month", y = "Mean Monthly Percentage Cancelled", fill = "Year") +
  scale_x_continuous(breaks = 1:12)
```



We can see high cancellation rates in the months of April, May and June on 2020 because of the lockdown for the pandemic.

We can also see high cancellation rates: - For 2018 in the months of April, May and June - In July and December for 2019 - In June and December 2022 These high cancellation rates are due to striking by SNCF employees against government plans to change the pension age.

Check the service variable.

Hide

```
df %>%
group_by(year,service) %>%
count()+
 geom_col(position = "stack", na.rm = TRUE)
```

```
Warning: Incompatible methods ("Ops.data.frame", "+.gg") for "+"Error in df %>% group_by(year, service) %>% count() + geom_c
ol(position = "stack",  :
  non-numeric argument to binary operator
```

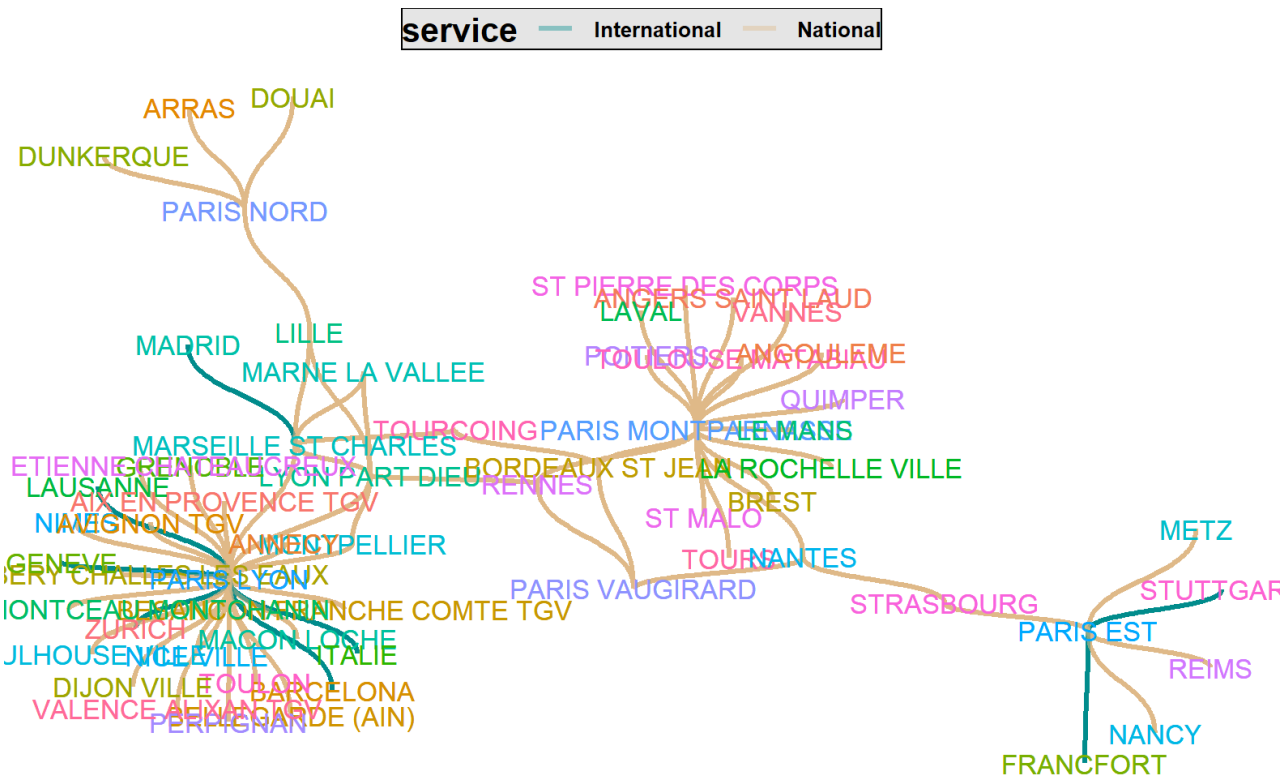
All years show similar number for the service.

We can use the ggraph package to plot the stations network.

Hide

```
p1 =  df %>%
  group_by(departure_station,arrival_station,service) %>%
  mutate(id = map2_chr(departure_station, arrival_station,
                       ~str_flatten(sort(c(.x,.y)))))%>%     # For each unique pairs of stations create an id
  group_by(id) %>%
  rename(from = departure_station,to = arrival_station)


p1 = p1[,c("from","to","service","id")]
graph1 = graph.data.frame(p1)



   ggraph(graph1) +
    geom_node_text(aes(label = name,color = name))+
  geom_edge_diagonal(aes(color = service),alpha = 0.4,width = 1)+
  scale_edge_color_manual(values=c("darkcyan","burlywood"))+
  geom_node_text(aes(label = name,color = name))+
    theme_void()+

    scale_colour_discrete(name  ="service",
                          breaks=c("National", "International"),
                          labels=c("National", "International"))+
    theme(legend.position="top",
          legend.background = element_rect(fill="gray90"),
        legend.text =element_text(face="bold"),
        legend.title = element_text(size=14,face="bold"))
```

```
Using "stress" as default layout
```



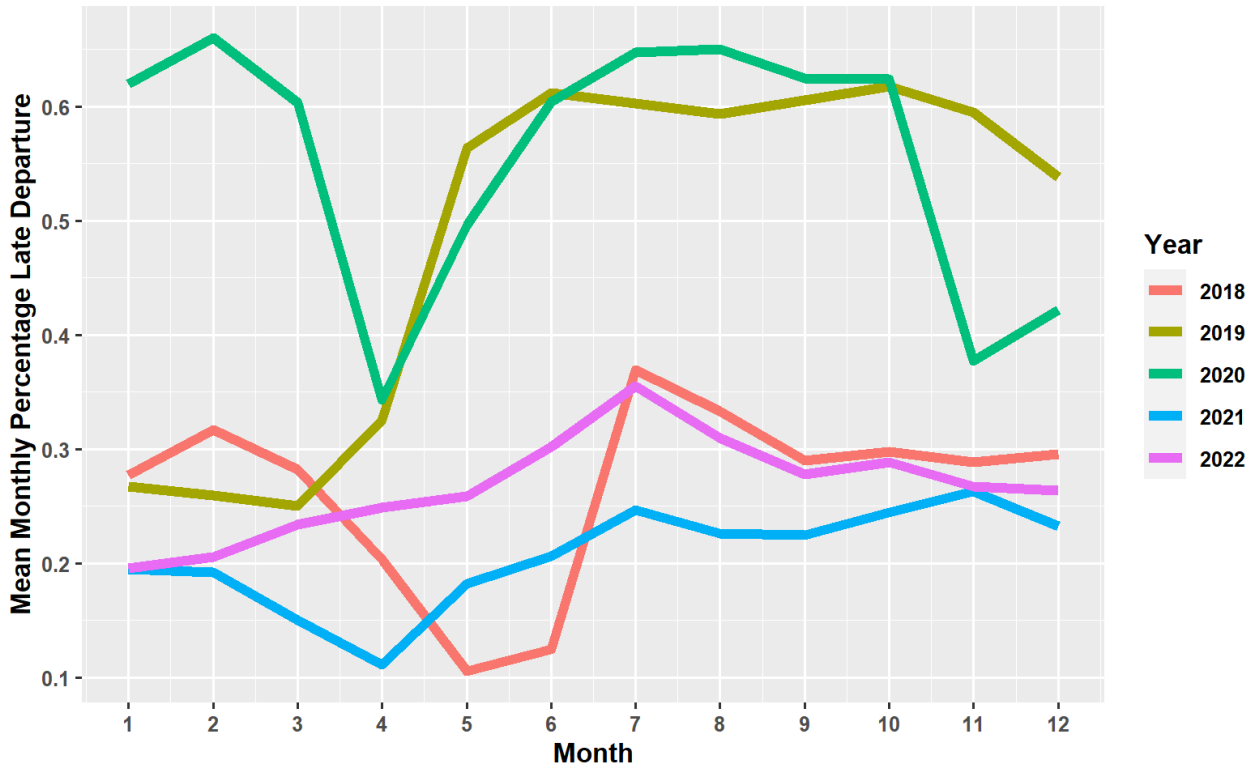
There are 6 international destinations: Italy,Frankfurt,Stuttgart,Lausanne,Zurich,Geneva. The two central stations are PARIS LYON and PARIS MONTPARNASSE.

Plot the mean monthly % of late departures for each year

Hide

```
df %>%
  mutate(pct_late_departure = num_late_at_departure/total_num_trips) %>%
  group_by(year, month) %>%
  summarise(monthly_pct_late_departure = mean(pct_late_departure, na.rm = TRUE)) %>%
  ggplot(aes(month, monthly_pct_late_departure, color = factor(year))) +
  geom_line(size = 2) +
  scale_x_continuous(breaks = 1:12) +
  labs(x = "Month", y = "Mean Monthly Percentage Late Departure", color = "Year") +
  theme(axis.text = element_text(face = "bold"),
        axis.title = element_text(face = "bold"),
        legend.text = element_text(face = "bold"),
        legend.title = element_text(face = "bold"))
```

```
`summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```



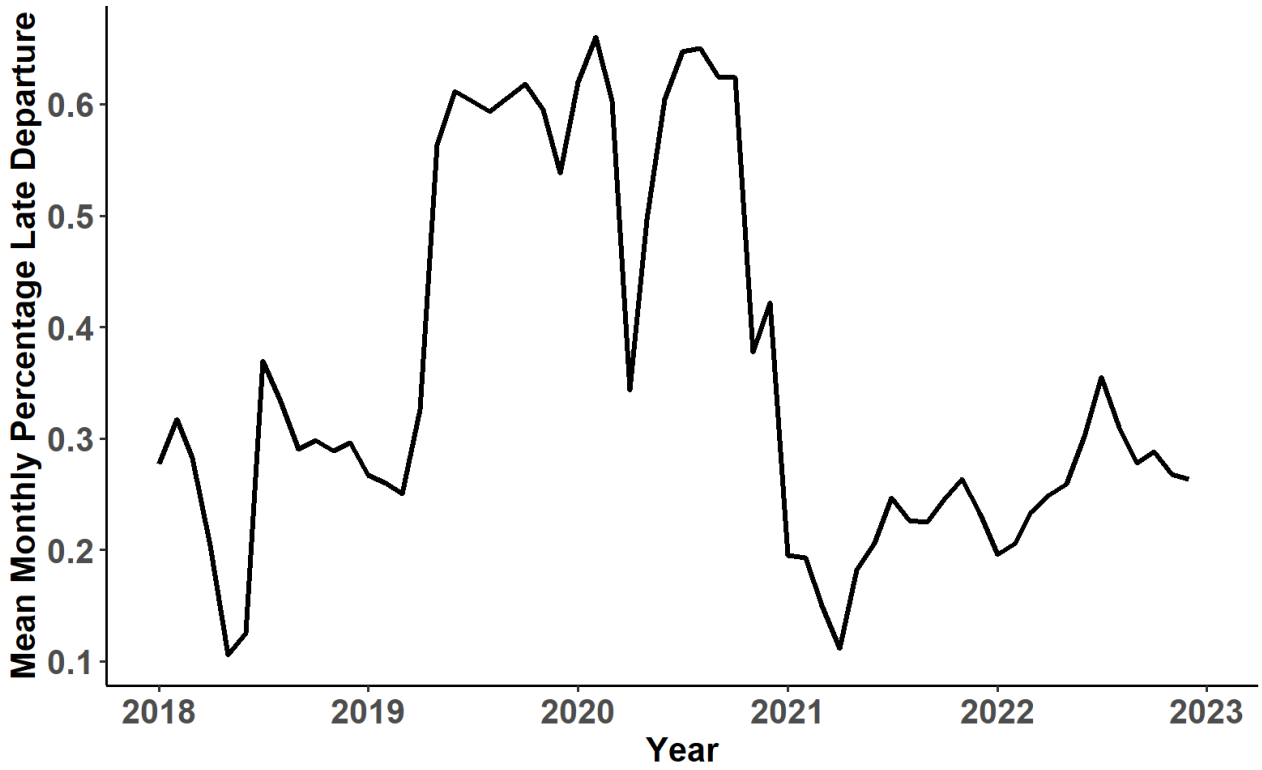
2019 and 2020 are showing excessively high percentage of late departures, around 60% on average for both years. The pandemic could be a reason for the high percentage in 2020. In 2019 the % of late departures increased dramatically after April, this could again be due to staff striking.

For 2018, 2021 and 2022, we see that the peak of delays occurs around June/July. This is the time when many people take holidays, so it could be due to that.

But that's quite a messy graph because we need to recode our year variable

Hide

```
df %>%
  group_by(date) %>%
  summarise(mean_monthly_pct_late_departure = mean(pct_late_departure, na.rm = TRUE)) %>%
  ggplot(aes(date, mean_monthly_pct_late_departure, size = 1)) +
  theme_classic() +
  geom_line(size = 1) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y", limits = as.Date(c("2018-01-01", "2022-12-31"))) +
  xlab("Year") +
  ylab("Mean Monthly Percentage Late Departure") +
  theme(axis.text = element_text(size = 14, face = "bold"),
        axis.title = element_text(size = 14, face = "bold"),
        legend.text = element_text(size = 14, face = "bold"),
        legend.title = element_text(size = 14, face = "bold"))
```



Let's compare cancellations and delays

Hide

```
p3 = df %>%
  group_by(year,date) %>%
  summarise(delays_rate = mean(pct_late_departure)) %>%
  ungroup() %>%
  ggplot(aes(date,delays_rate,fill=delays_rate))+
  scale_fill_gradient(low="pink", high="red")+
  scale_y_continuous(labels = percent_format())+
  geom_bar(stat = "identity")+
theme(axis.text=element_text(size=16,face="bold"),
        axis.title=element_text(size=16,face="bold"),
        legend.text =element_text(size=16,face="bold"),
        legend.title = element_text(size=16,face="bold"))
```

```
`summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

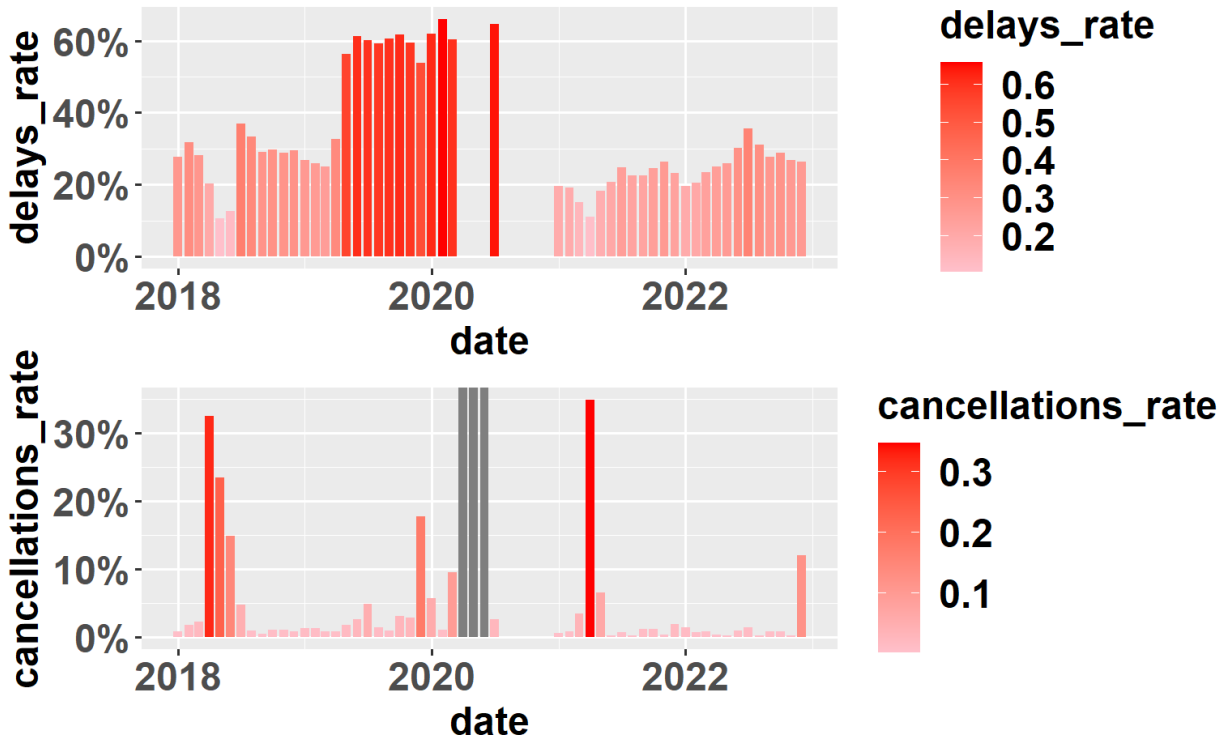
Hide

```
p4 = df %>%
  group_by(year,date) %>%
  summarise(cancellations_rate = mean(mean_monthly_pct_cancelled)) %>%
  ggplot(aes(date,cancellations_rate,fill=cancellations_rate))+
  scale_fill_gradient(low="pink", high="red")+
  scale_y_continuous(labels = percent_format())+
  geom_bar(stat = "identity")+
theme(axis.text=element_text(size=16,face="bold"),
        axis.title=element_text(size=16,face="bold"),
        legend.text =element_text(size=16,face="bold"),
        legend.title = element_text(size=16,face="bold"))
```

```
`summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

Hide

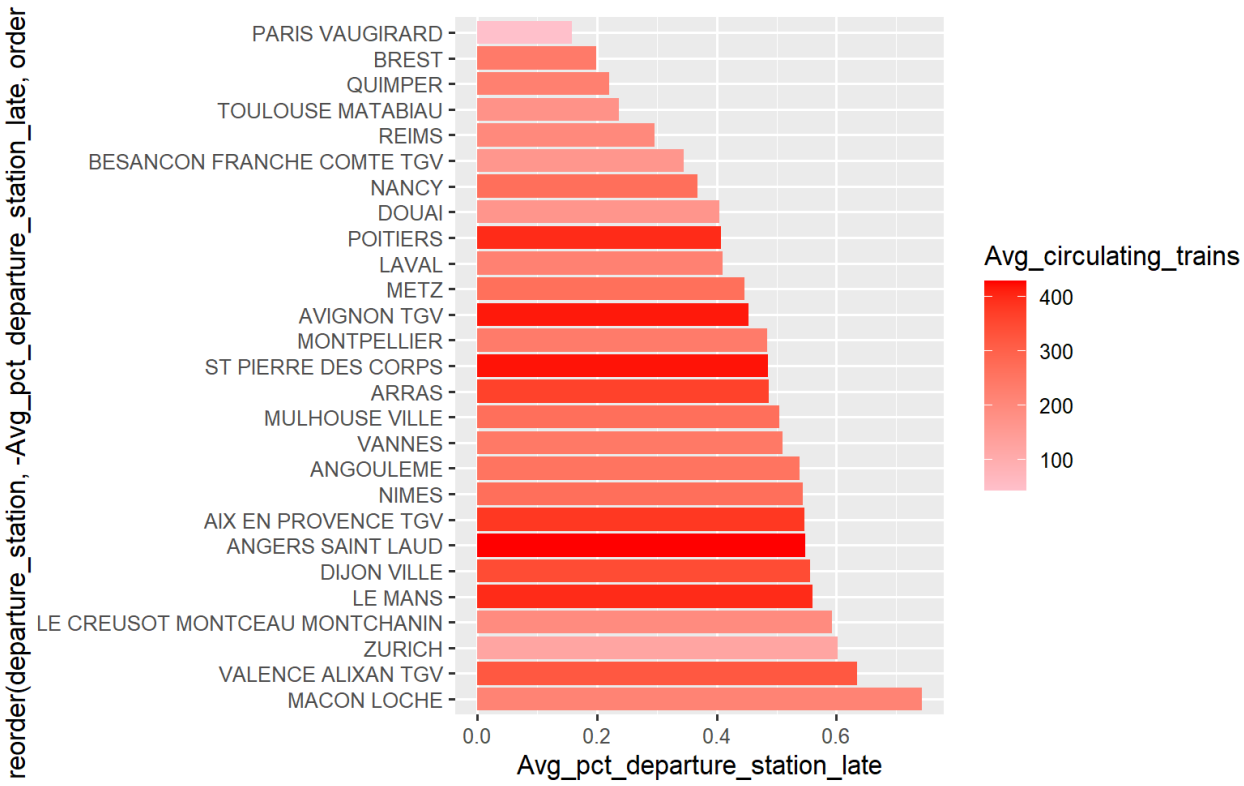
```
p3/p4
```



Check if busy stations with more trains tend to have delays more than less busy stations.

Hide

```
df %>%
  group_by(departure_station) %>%
  summarise(Avg_pct_departure_station_late = mean(pct_late_departure),
            Avg_circulating_trains = mean(total_num_trips)) %>%
  arrange(desc(Avg_pct_departure_station_late)) %>%
  filter(Avg_pct_departure_station_late>0.1) %>%
  ggplot(aes(x = reorder(departure_station, -Avg_pct_departure_station_late,order = F), y= Avg_pct_departure_station_late,fi
ll= Avg_circulating_trains))+
  scale_fill_gradient(low='pink', high='red')+
  geom_bar(stat = "identity")+
  coord_flip()
```

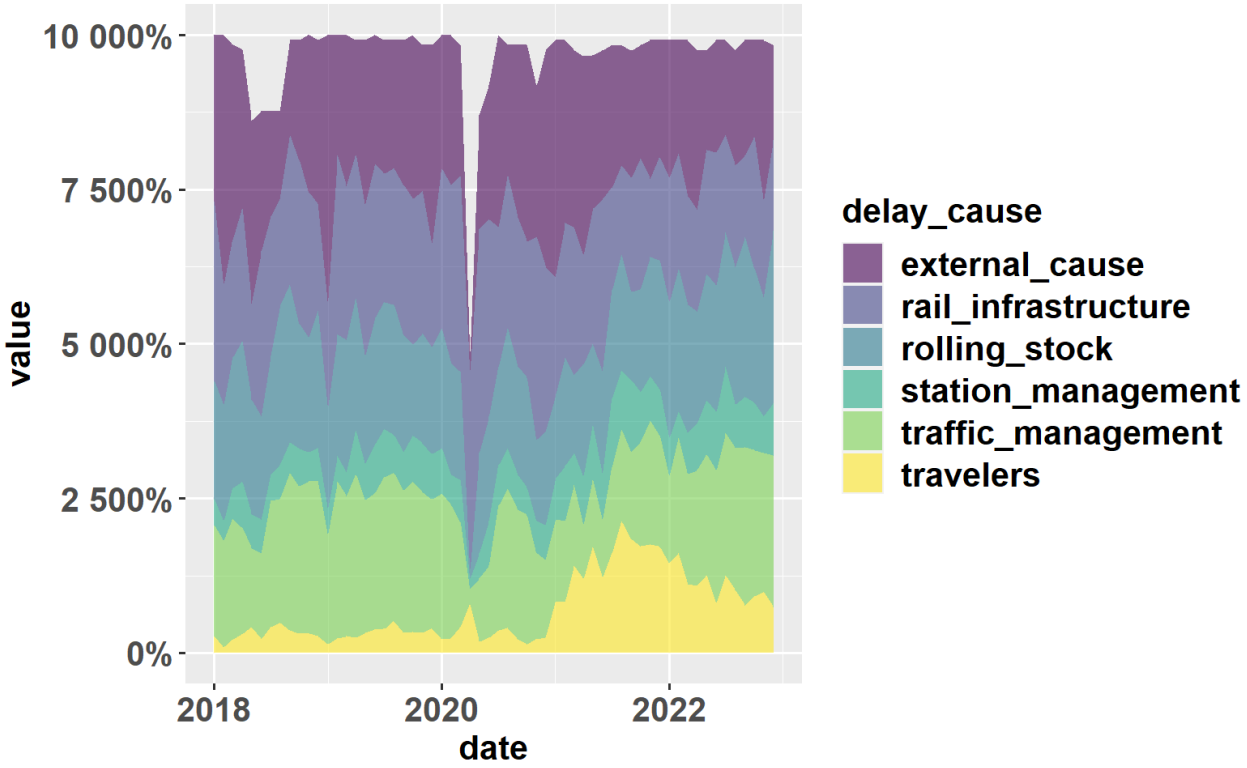


How do the causes of delay fluctuate over time?

Hide

```
df %>%
  group_by(year,month) %>%
        summarise(across(starts_with("delay_"), mean,na.rm = T)) %>%
         mutate(date = as.Date(sprintf("%d-%02d-01",year,month))) %>%
        pivot_longer(cols = starts_with("delay"),names_to = "delay") %>%
        mutate(delay = substring(delay,nchar("delay_cause_")+1)) %>%
        rename(delay_cause = delay) %>%
  ggplot(aes(date,value,fill = delay_cause))+
  scale_fill_viridis(discrete = T)+
  geom_area(alpha = 0.6)+
  scale_y_continuous(labels = percent_format())+
theme(axis.text=element_text(size=14,face="bold"),
        axis.title=element_text(size=14,face="bold"),
        legend.text =element_text(size=14,face="bold"),
        legend.title = element_text(size=14,face="bold"))
```

```
`summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

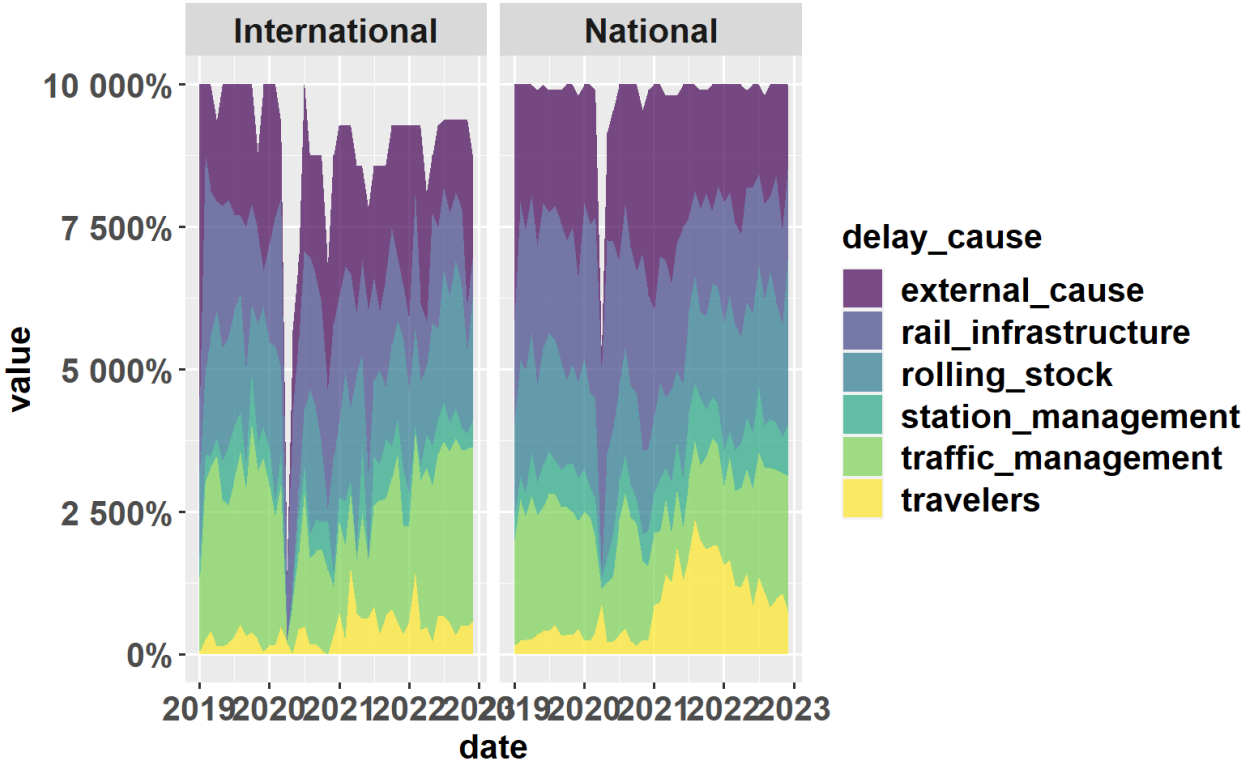


Group by service

Hide

```
df %>%
  filter(year != 2018) %>%
  group_by(year,month,service) %>%
        summarise(across(starts_with("delay_"), mean,na.rm = T)) %>%
         mutate(date = as.Date(sprintf("%d-%02d-01",year,month))) %>%
        pivot_longer(cols = starts_with("delay"),names_to = "delay") %>%
        mutate(delay = substring(delay,nchar("delay_cause_")+1)) %>%
        rename(delay_cause = delay) %>%
  ggplot(aes(date,value,fill = delay_cause))+
  scale_fill_viridis(discrete = T)+
  facet_grid(~service) +
  geom_area(alpha = 0.7)+
  scale_y_continuous(labels = percent_format())+
theme(axis.text=element_text(size=14,face="bold"),
        axis.title=element_text(size=14,face="bold"),
        legend.text =element_text(size=14,face="bold"),
        legend.title = element_text(size=14,face="bold"),
    strip.text = element_text(face="bold", size=14))
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the `.groups` argument.
```


International
National
10 000%
7 500%
5 000%
2 500%
0%
value
2019
2020
2021
2022
2023
date
delay_cause
external_cause
rail_infrastructure
rolling_stock
station_management
traffic_management
travelers