Assignment 1:“.

David

2023-05-17

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#libraries
library('data.table')
library('ggplot2')
library('dplyr')
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

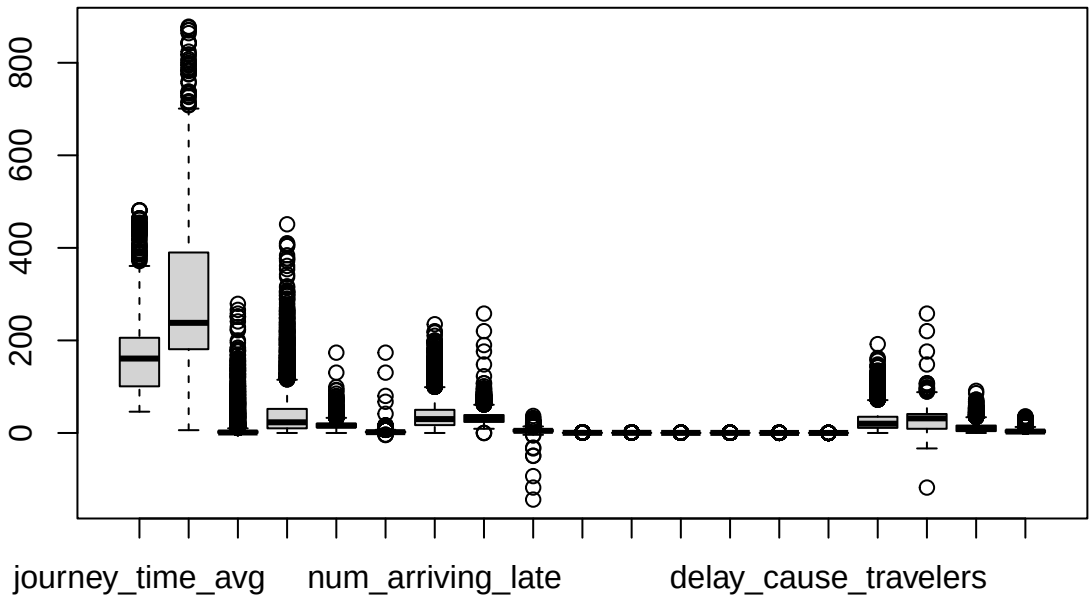
```
library('tidyr')
library('stats')

#Dataset
dataset <- read.csv("/home/dennis/Desktop/Data-Science-analytics-R/david/dataset/full_trains.csv", head

#theme for plots
theme_set(theme_classic())
theme_update(plot.title = element_text((hjust = 0.5)))

#EDA
numeric_cols <- sapply(dataset,is.numeric)
numeric_dataset <- dataset[,numeric_cols]

#futher EDA
numeric_dataset <- numeric_dataset[, !colnames(numeric_dataset) %in% 'year']
numeric_dataset <- numeric_dataset[, !colnames(numeric_dataset) %in% 'month']
boxplot(numeric_dataset)
```




```
#preprocessing
#removing NA
#which(is.na(dataset))

dataset <- subset.data.frame(dataset,select = -c(comment_cancellations,comment_delays_on_arrival,commen
dataset[dataset < 0] <- NA
dataset <- drop_na(dataset)

#investigating factors causing variance in trip time
cancellations <- dataset %>% group_by(departure_station) %>% summarise(cancelled_trips= sum(num_of_canc

# Calculate total number of delays caused by external factors for each train station
delays_external<- dataset %>% group_by(departure_station ) %>% summarise(external_f = sum(delay_cause_e
cancellations <- merge(cancellations, delays_external, by = "departure_station")
# Sort train stations by external delays in descending order
stations_sorted <- cancellations[order(cancellations$external_f, decreasing = TRUE), ]
# Explore train stations with higher cancellation rates and their external factors
head(stations_sorted)
```

```
##       departure_station cancelled_trips external_f
## 36           PARIS LYON            1955  226.35155
## 37   PARIS MONTPARNASSE            1823  160.55625
## 35            PARIS EST             480   51.46250
## 25       LYON PART DIEU             747   49.61424
## 38           PARIS NORD             323   42.62293
## 27 MARSEILLE ST CHARLES             450   31.73556
```

```
# Calculate total number of delays caused by management for each train station
delay_management<- dataset %>% group_by(departure_station) %>% summarise(delays_management = sum(delay_
# Merge with cancellations dataset
cancellations <- merge(cancellations, delay_management, by = "departure_station")
# Sort train stations by external delays in descending order
stations_sorted <- cancellations[order(cancellations$delays_management, decreasing = TRUE), ]
# Explore train stations with higher cancellation rates and their external factors
head(stations_sorted)
```

```
##       departure_station cancelled_trips external_f delays_management
## 36           PARIS LYON            1955  226.35155          59.75320
## 37   PARIS MONTPARNASSE            1823  160.55625          52.52834
## 35            PARIS EST             480   51.46250          20.83542
## 38           PARIS NORD             323   42.62293          19.00973
## 25       LYON PART DIEU             747   49.61424          15.22221
## 27 MARSEILLE ST CHARLES             450   31.73556           9.75600
```

```
# Calculate total number of delays caused by rail infrastructure for each train station
rail_infra_delays <- dataset %>% group_by(departure_station) %>% summarise(rail_infra_delays = sum(delay
cancellations <- merge(cancellations, rail_infra_delays, by = "departure_station")
# Sort train stations by total rail infrastructure delays in descending order
stations_sorted <- cancellations[order(cancellations$rail_infra_delays, decreasing = TRUE), ]
# Explore train stations with higher cancellation rates and their rail infrastructure delays
head(stations_sorted)
```

```
##     departure_station cancelled_trips external_f delays_management
## 37 PARIS MONTPARNASSE            1823  160.55625          52.52834
## 36         PARIS LYON            1955  226.35155          59.75320
## 35          PARIS EST             480   51.46250          20.83542
## 25     LYON PART DIEU             747   49.61424          15.22221
## 38         PARIS NORD             323   42.62293          19.00973
## 32             NANTES             240   20.81057           3.88123
##    rail_infra_delays
## 37         176.73073
## 36         156.89264
## 35          41.73882
## 25          41.54865
## 38          26.57829
## 32          24.80821
```

```
# Calculate total number of delays caused by rail infrastructure for each train station
rail_infra_delays <- dataset %>% group_by(departure_station) %>% summarise(travelers_delays = sum(delay
cancellations <- merge(cancellations, rail_infra_delays, by = "departure_station")
# Sort train stations by total rail infrastructure delays in descending order
stations_sorted <- cancellations[order(cancellations$travelers_delays, decreasing = TRUE), ]
# Explore train stations with higher cancellation rates and their rail infrastructure delays
head(stations_sorted)
```

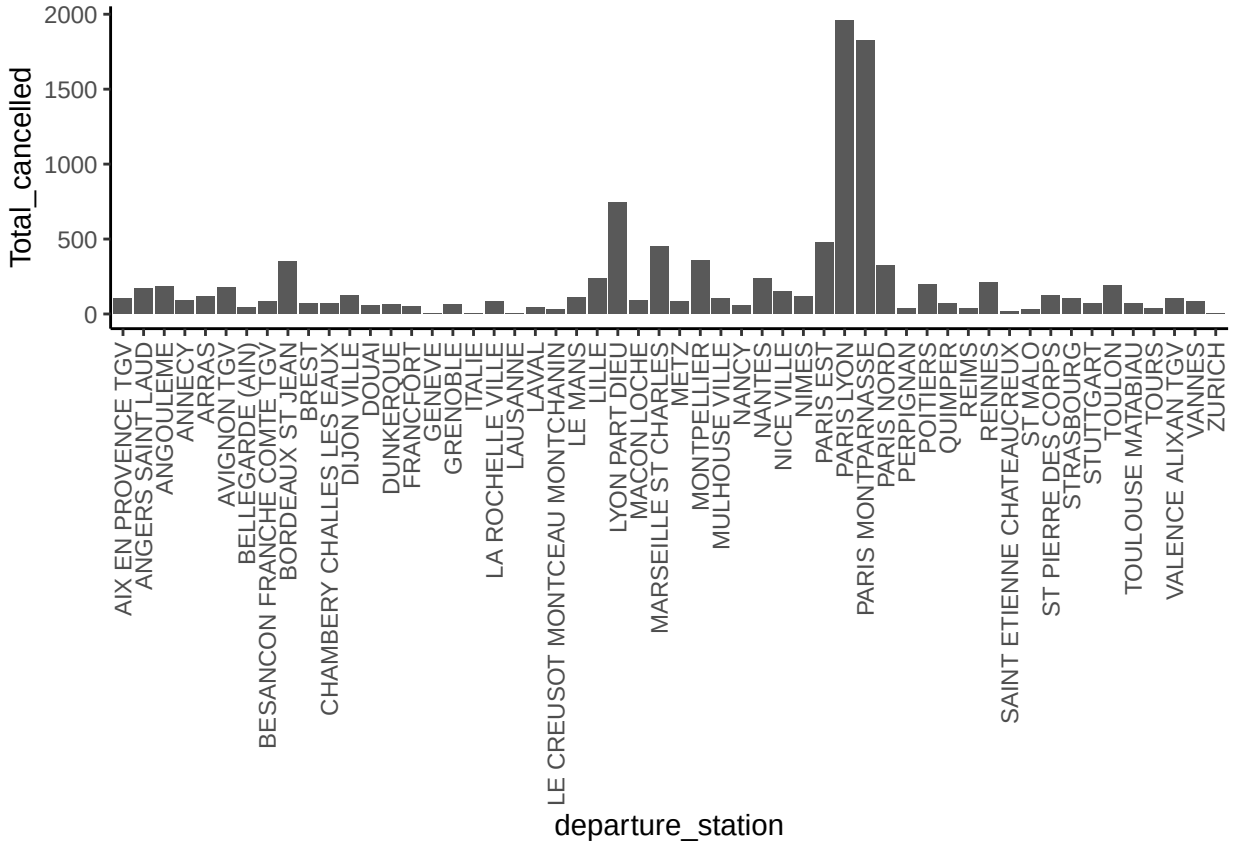
```
##     departure_station cancelled_trips external_f delays_management
## 36         PARIS LYON            1955  226.35155         59.753203
## 37 PARIS MONTPARNASSE            1823  160.55625         52.528341
## 25     LYON PART DIEU             747   49.61424         15.222209
## 38         PARIS NORD             323   42.62293         19.009732
## 24              LILLE             240   26.33584          8.372758
## 35          PARIS EST             480   51.46250         20.835424
##    rail_infra_delays travelers_delays
## 36         156.89264        25.642206
## 37         176.73073        17.733152
## 25          41.54865        10.054316
## 38          26.57829         6.732791
## 24          18.59446         6.510113
## 35          41.73882         6.125741
```

```
#Calculate total number of delays caused by rail rolling stock for each train station
rolling_stock_delays <- dataset %>% group_by(departure_station) %>% summarise(rolling_stock_delay = sum
```

```
cancellations <- merge(cancellations, rolling_stock_delays, by = "departure_station")
# Sort train stations by total rail infrastructure delays in descending order
stations_sorted <- cancellations[order(cancellations$rolling_stock_delay, decreasing = TRUE), ]
# Explore train stations with higher cancellation rates and their rail infrastructure delays
head(stations_sorted)
```

```
##     departure_station cancelled_trips external_f delays_management
## 36         PARIS LYON            1955  226.35155         59.753203
## 37 PARIS MONTPARNASSE            1823  160.55625         52.528341
## 35          PARIS EST             480   51.46250         20.835424
## 25     LYON PART DIEU             747   49.61424         15.222209
## 38         PARIS NORD             323   42.62293         19.009732
## 24              LILLE             240   26.33584          8.372758
##    rail_infra_delays travelers_delays rolling_stock_delay
## 36         156.89264        25.642206           173.89533
## 37         176.73073        17.733152            92.83053
## 35          41.73882         6.125741            38.52222
## 25          41.54865        10.054316            28.19754
## 38          26.57829         6.732791            22.18322
## 24          18.59446         6.510113            18.83090
```

```
#question 1
#Distribution of cancelled trains
distribution <- dataset %>% group_by(departure_station) %>% summarise(Total_cancelled = sum(num_of_canc
ggplot(data = distribution, aes(x= departure_station,y=Total_cancelled))+ geom_bar(stat = 'identity')+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```

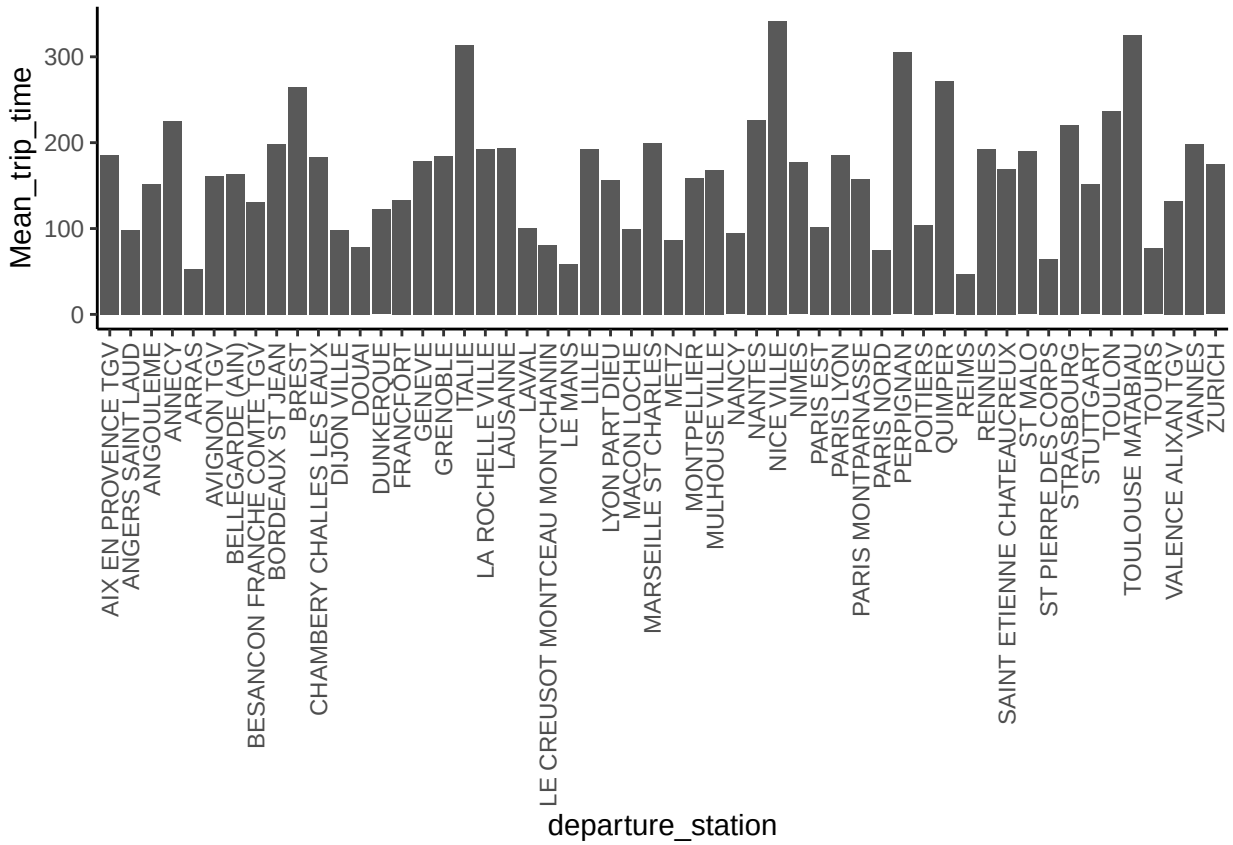


```r
#comparing cancellations with the various factors
anova_test <- aov(cancelled_trips ~ rail_infra_delays + external_f + delays_management + rolling_stock_
summary(anova_test)
```

```
##                      Df  Sum Sq Mean Sq  F value   Pr(>F)
## rail_infra_delays     1 6557874 6557874 1152.600  < 2e-16 ***
## external_f            1   88988   88988   15.640 0.000246 ***
## delays_management     1    1091    1091    0.192 0.663377
## rolling_stock_delay   1    8240    8240    1.448 0.234587
## Residuals            49  278792    5690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#conclusion
#external factors and rail infrastructure delays are the most significant factors affecting cancellatio


#quetion 2
#Average trip times
average_trip_time <- dataset %>% group_by(departure_station) %>% summarise(Mean_trip_time = mean(journey
ggplot(data = average_trip_time, aes(x= departure_station,y=Mean_trip_time))+ geom_bar(stat = 'identity
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```




```r
#comparing trip time with various factors causing delays
cancellations<-merge(cancellations,average_trip_time,by = 'departure_station')
stations_sorted <- cancellations[order(cancellations$Mean_trip_time, decreasing = TRUE), ]

anova_test <- aov(Mean_trip_time ~ rail_infra_delays + external_f + delays_management + rolling_stock_d
```

```r
summary(anova_test)
```

```
##                      Df Sum Sq Mean Sq F value   Pr(>F)
## rail_infra_delays     1     20      20   0.006 0.939777
## external_f            1   6068    6068   1.745 0.192676
## delays_management     1  57446   57446  16.518 0.000174 ***
## rolling_stock_delay   1  30643   30643   8.811 0.004623 **
## Residuals            49 170412    3478
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#conclusions
#delays caused by rolling stock and management have the greatest impact on average_trip_time in the
#stations


#trips by month
varying_trips <- dataset %>% group_by(year,month,departure_station) %>% summarise(trips_by_station = su
```

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

```r
varying_trips$Date <- as.Date(paste0(varying_trips$year, "-", varying_trips$month, "-01"))
varying_trips <- subset.data.frame(varying_trips,select = -c(year,month))

which.max(varying_trips$trips_by_station)
```

```
## [1] 1638
```

```r
which.min(varying_trips$trips_by_station)
```

```
## [1] 1204
```

```r
mean(varying_trips$trips_by_station)
```

```
## [1] 591.0145
```

```r
varying_trips <- varying_trips %>%group_by(departure_station) %>% summarise(max_trips = max(trips_by_st
varying_trips <- varying_trips[order(varying_trips$max_trips, decreasing = TRUE), ]
head(varying_trips)
```

```
## # A tibble: 6 x 2
##   departure_station    max_trips
##   <chr>                    <int>
## 1 PARIS LYON                6623
## 2 PARIS MONTPARNASSE        5743
## 3 LYON PART DIEU            1993
## 4 PARIS EST                 1650
## 5 PARIS NORD                1491
## 6 MARSEILLE ST CHARLES      1258
```

```r
tail(varying_trips)
```

```
## # A tibble: 6 x 2
##   departure_station          max_trips
##   <chr>                          <int>
## 1 TOULOUSE MATABIAU                191
## 2 FRANCFORT                        177
```

```
## 3 LAUSANNE                          146
## 4 ST MALO                           121
## 5 SAINT ETIENNE CHATEAUCREUX        119
## 6 ITALIE                            113
```

```
#conclusion
#From the Analysis PARIS LYON AND PARIS MONTPARNASSE  seem to be the best performing stations
#by trips. however the high volume of traffic
#seems to come with high rate of cancellations and high delay times
```