

# TOTVS\_DataChallenge

*Mauricio Noris Freire*

*October 26, 2016*

The data available goes back three weeks and based on the amount expended every day, a forecast was made to estimate the revenue for the next week.

```
##Getting the data

setwd("E:/TOTVS_DataChallenge")
data <- 'data/sample.txt'

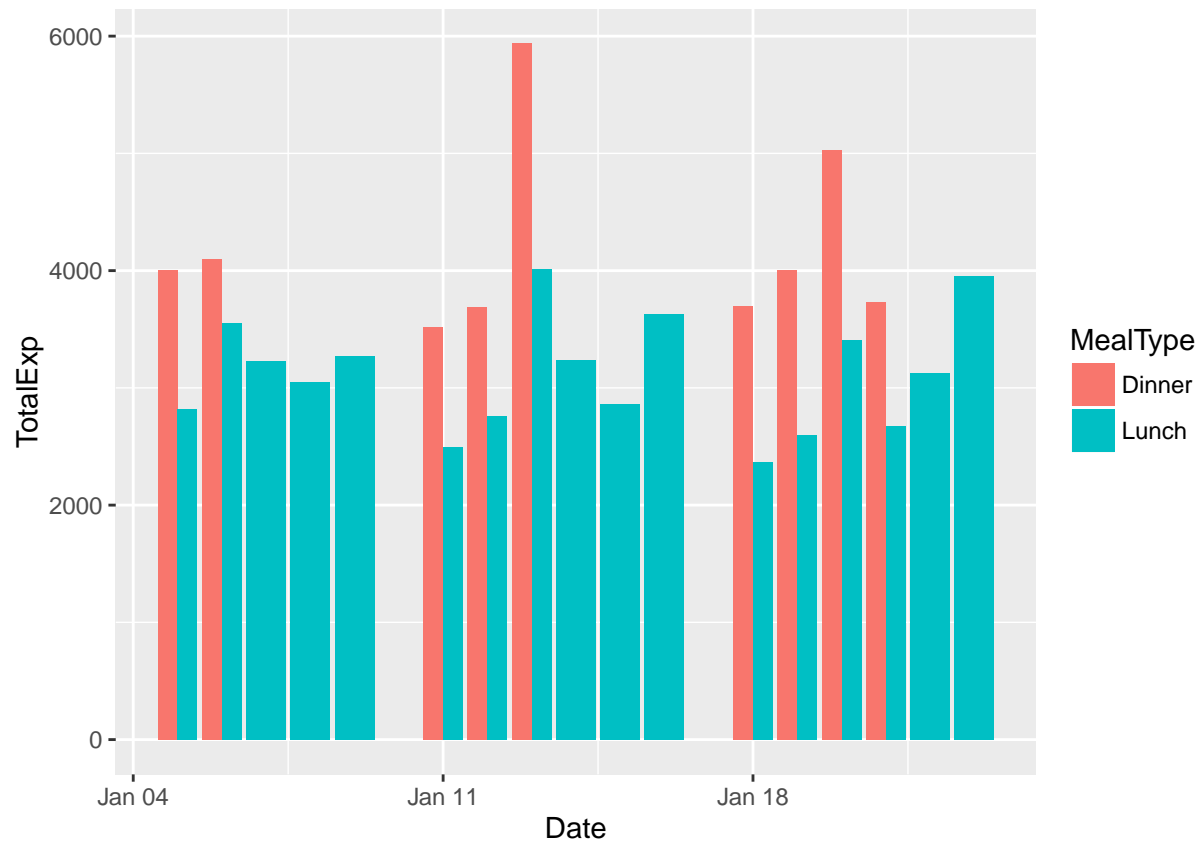
df <- fromJSON(data)
df$dt <- strptime(df$idhEmi$date, "%Y-%m-%dT%H:%M:%S", tz="Zulu")

df$mealtype = NA
df[hour(df$dt) >= 11 & hour(df$dt) < 17,]$mealtype <- 'Lunch'
df[hour(df$dt) >= 17,]$mealtype <- 'Dinner'

#consolidate
df.exp.daily <- aggregate(df$complemento$valorTotal ~ as.Date(df$dt)+df$mealtype, df, sum)
colnames(df.exp.daily) <- c('Date', 'MealType', 'TotalExp')
df.exp.daily$yday <- strftime(df.exp.daily$Date, '%A')
df.exp.daily$wnum <- strftime(df.exp.daily$Date, '%W')
```

The following plot shows the total amount expended from 2016-01-05 and 2016-01-23 splited between the Lunch and Dinner.

```
ggplot(df.exp.daily, aes(x = Date, y = TotalExp, fill=MealType)) + geom_bar(stat="identity", position="stack")
```

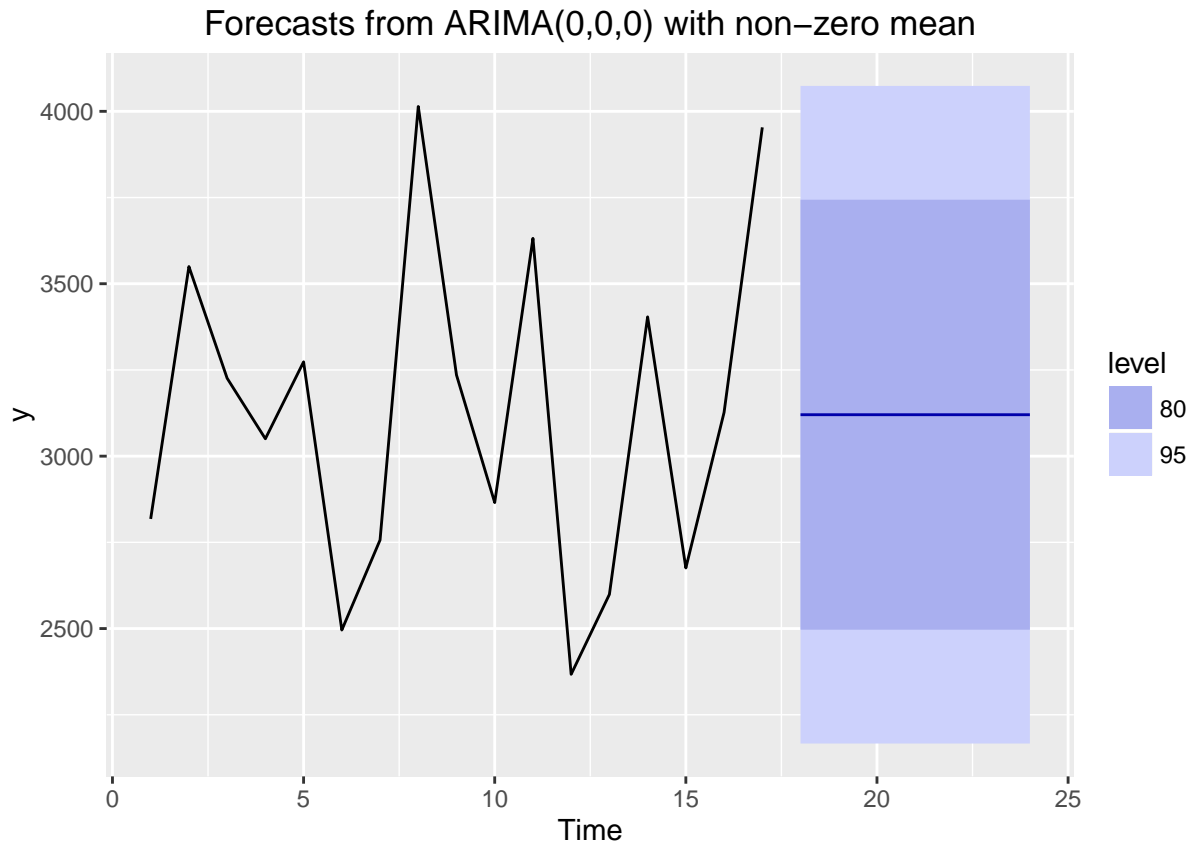


There is no data for dinner every day and, because of that, two forecasts were created. The first dataset removes data from dinners and a simulation was made based on the hypothesis of the restaurant serving only Lunch. The following plot exhibits the result of that simulation. It was used the ARIMA model because it is a short term forecast.

The expected daily revenue for lunch is:

- 1) 80% of confidence between \$2496.641 and \$3743.642
- 2) 95% of confidence between \$2166.58 and \$4073.703

```
#only lunch
df.exp.daily.lunch <- df.exp.daily[df.exp.daily$MealType == 'Lunch',]
df.exp.daily.lunch$TotalExp <- ts(df.exp.daily.lunch$TotalExp,frequency=1)
fit <- auto.arima(df.exp.daily.lunch$TotalExp)
f<-forecast(fit,h=7)
autoplot(f)
```



The second forecast puts together lunches and dinners. The expected daily revenue is:

- 1) 80% of confidence between \$ 2516.666 and \$ 8159.725
- 2) 95% of confidence between \$ 1023.038 and \$ 9653.353

```
#full
df.exp.daily.full <- aggregate(df.exp.daily$TotalExp ~ as.Date(df.exp.daily$Date), df.exp.daily, sum)
colnames(df.exp.daily.full) <- c('Date', 'TotalExp')
df.exp.daily.full$TotalExp <- ts(df.exp.daily.full$TotalExp, frequency=1)
fit1 <- auto.arima(df.exp.daily.full$TotalExp)
f1<-forecast(fit1,h=7)
autoplot(f1)
```

