Time Series Analysis

using Apple Mobility Trends Data

Eldhose Poulose

24.04.2021

Introduction

Through this RMarkdown I want to present my sample work in Time Series Analysis, which I have done using the Apple Mobility Trends Data. I found pulling down this file challenging due to the fact that the URL for the CSV file changes daily. I mainly used the "rvest" package to harvest data from html pages and in python I use "beautifulSoup" package to scrape data. Eventhough I have experience in both I found this task challenging because my goal was to automate the entire process by harvesting the data from this page every day, run the job and mail the results to my E-Mail. I found that the URL for dataset is dynamic in some random way or as a function of the version (also dynamic) of the web content management system. Thus I understood that I can't easily scrape it and just look for the URL embedded in the "Download the Data" button. After couple of days of research I found that the index.json contains the stable/URL of the dataset for each day.

The reason behind selecting this data is becasuse of the importance of this data in this period of Pandemic. I want to explore the activities of the public and see how it is changing everyday.

Steps

- Ingest Data
  - Apple Mobility Trends Data
- Merge datasets
- Exploratory Data Analysis and Visualizations
- Time Series Analysis and Forecasts
- Do "out of the box" time series forecast.
- Analyze fluctuations around time series trends.

Remark: The time series section is done for illustration purposes only. The forecasts there should not be taken seriously.

Motivation

Overall Process

Dataset

Apple Mobility Trends Data

- Download the data The Apple's page provides the Dataset for this data analysis.
- Import the data and summarize it
- Transform the data into long form
- Partition the data into subsets that correspond to combinations of geographical regions and transportation types
- Make contingency matrices and corresponding heat-map plots
- Make NN graphs over the contingency matrices and plot communities
- Plot the corresponding Time Series

Ingest Data

```
get_apple_target <- function(cdn_url = "https://covid19-static.cdn-apple.com",
                             json_file = "covid19-mobility-data/current/v3/index.json") {
  tf <- tempfile(fileext = ".json")
  curl::curl_download(paste0(cdn_url, "/", json_file), tf)
  json_data <- jsonlite::fromJSON(tf)
  paste0(cdn_url, json_data$basePath, json_data$regions$`en-us`$csvPath)
}
get_apple_target()
```

```
## [1] "https://covid19-static.cdn-apple.com/covid19-mobility-data/2109HotfixDev27/v3/en-us/applemobili
```

```
fileName <- tail(unlist(strsplit(get_apple_target(),"/")), n=1)
download.file(get_apple_target(),fileName)
appleMobility <- read.csv(fileName,stringsAsFactors = TRUE)
```

Observation: Reference date for normalization is January 13, 2020. Note the values in that column are set to 100.

Data Dimension

```
dim(appleMobility)
```

```
## [1] 4691  526
```

```
summary(as.data.frame(unclass(appleMobility[,1:3]), stringsAsFactors = TRUE))
```

```
##            geo_type                  region     transportation_type
##  city          : 790   Washington County:  27   driving:3048
##  country/region: 153   Jefferson County :  25   transit: 551
##  county        :2638   Montgomery County:  24   walking:1092
##  sub-region    :1110   Franklin County  :  22
##                        Madison County   :  21
##                        Jackson County   :  19
##                        (Other)          :4553
```

Answering basic questions about the Dataset

- Here I try to answer the basic questions which I come across from the data.
  - How many geo_types are present in the data?
  - How many unique regions are present in the data?
  - How many transportation modes are present in the data?
  - How many Countries are present in the data? **Note:** Big countries are divides into regions are provided them in the regions column.

```
lsGeo_type <- as.String(levels(appleMobility$geo_type))
lsGeo_type
```

```
## city
## country/region
## county
## sub-region
```

```
print(length(levels(appleMobility$region)))
```

```
## [1] 2325
```

```
#lsRegion <- as.String(levels(appleMobility$region))
#lsRegion
```

```
print(as.String(levels(appleMobility$transportation_type)))
```

```
## driving
## transit
## walking
```

```
print(length(levels(appleMobility$country)))
```

```
## [1] 48
```

```
#print(as.String(levels(appleMobility$country)))
```

Data transformation

Here I convert the data into narrow/long format. For this I use melt function from reshape2 package.

```
#appleMobility_melted <- melt(appleMobility,id=c("geo_type","region","transportation_type","alternative
colNames <- c("geo_type","region","transportation_type","alternative_name","sub.region","country")
appleMobility_melted <- tidyr::pivot_longer( data = appleMobility, cols = setdiff( names(appleMobility)
```

Remove empty rows

```
appleMobility_na <- appleMobility_melted[complete.cases(appleMobility_melted), ]
```

Clean Date Format

```
appleMobility_na$Date <- gsub("[a-zA-Z ]", "", appleMobility_na$Date)
```

Add: DateInfo, Day

```
appleMobility_na$Date <- as.POSIXct(appleMobility_na$Date, format = "%Y.%m.%d", origin = "1970.01.01" )
appleMobility_na$Day <- weekdays(appleMobility_na$Date)
appleMobility_na$Day <- as.factor(appleMobility_na$Day)
appleMobility_na$DateInfo <- format(appleMobility_na$Date, "%a %b %d %Y")
appleMobility_na$DateInfo <- as.factor(appleMobility_na$DateInfo)
```

Draw Random Samples

```
set.seed(123)
appleMobility_na %>% dplyr::sample_n(10)
```

```
## # A tibble: 10 x 10
##    geo_type region transportation_~ alternative_name sub.region country
##    <fct>    <fct>  <fct>            <fct>            <fct>      <fct>
##  1 county   Picke~ walking          ""               "South Ca~ United~
##  2 city     Brisb~ transit          ""               "Queensla~ Austra~
##  3 county   Klama~ driving          ""               "Oregon"   United~
##  4 county   Mono ~ driving          ""               "Californ~ United~
##  5 sub-reg~ Alber~ walking          "AB"             ""         Canada
##  6 city     San F~ walking          ""               "Californ~ United~
##  7 sub-reg~ Catal~ transit          "Cataluña|Catal~ ""         Spain
##  8 county   Cataw~ driving          ""               "North Ca~ United~
##  9 sub-reg~ Rhode~ walking          ""               ""         United~
## 10 sub-reg~ São P~ driving          "São Paulo (Est~ ""         Brazil
## # ... with 4 more variables: Date <dttm>, value <dbl>, Day <fct>,
## #   DateInfo <fct>
```

##Summary of the cleaned data

```
summary(as.data.frame(unclass(appleMobility_na), stringsAsFactors = TRUE))
```

```
##            geo_type                      region        transportation_type
##  city          : 408438   Washington County:  13969   driving:1555653
##  country/region:  79101   Jefferson County :  12937   transit: 285525
##  county        :1364942   Montgomery County:  12426   walking: 566038
##  sub-region    : 554735   Franklin County  :  11378
##                           Madison County   :  10863
##                           Jackson County   :   9829
##                           (Other)          :2335814
##                 alternative_name        sub.region               country
##                         :1880288            : 685027   United States:1605018
##  AB                     :   1555   Texas     : 124673   Japan        : 114445
##  ACT                    :   1555   California:  85930                :  79101
##  Andalucía              :   1555   Georgia   :  67769   France       :  46598
##  Bayern                 :   1555   Virginia  :  63633   Germany      :  44508
##  BC|Colombie-Britannique:   1555   Florida   :  62643   Thailand     :  35168
##  (Other)                : 519153   (Other)   :1317541   (Other)      : 482378
```

```
##       Date                          value               Day
##  Min.   :2020-01-13 00:00:00   Min.   :   0.44   Friday   :339596
##  1st Qu.:2020-05-22 00:00:00   1st Qu.:  87.54   Monday   :345314
##  Median :2020-09-29 00:00:00   Median : 117.39   Saturday :344248
##  Mean   :2020-09-28 14:49:32   Mean   : 124.70   Sunday   :344248
##  3rd Qu.:2021-02-05 00:00:00   3rd Qu.: 152.33   Thursday :344248
##  Max.   :2021-06-15 00:00:00   Max.   :2148.12   Tuesday  :345314
##                                                  Wednesday:344248
##             DateInfo
##  Fri Apr 02 2021:   4652
##  Fri Apr 03 2020:   4652
##  Fri Apr 09 2021:   4652
##  Fri Apr 10 2020:   4652
##  Fri Apr 16 2021:   4652
##  Fri Apr 17 2020:   4652
##  (Other)        :2379304
```

Data Partition

*I am interested in the geographical types and transportation modes. Therefore we group the data as per this requirement.

```
appleMobility_na %>%
  dplyr::group_by(geo_type, transportation_type) %>%
  dplyr::count()
```

```
## # A tibble: 12 x 3
## # Groups:   geo_type, transportation_type [12]
##    geo_type       transportation_type       n
##    <fct>          <fct>                 <int>
##  1 city           driving              154583
##  2 city           transit              101853
##  3 city           walking              152002
##  4 country/region driving               32571
##  5 country/region transit               13959
##  6 country/region walking               32571
##  7 county         driving             1080530
##  8 county         transit               78888
##  9 county         walking              205524
## 10 sub-region     driving              287969
## 11 sub-region     transit               90825
## 12 sub-region     walking              175941
```

```
appleMobility_part <- split(appleMobility_na, appleMobility_na[,c("geo_type","transportation_type")])
```

Heat-Map Plots

Contigency Matrix

```
aMatDateRegion <- purrr::map(appleMobility_part, function(dfX) { xtabs( formula = value ~ Date + region
aMatDateRegion <- aMatDateRegion[ purrr::map_lgl(aMatDateRegion, function(x) nrow(x) > 0 ) ]
```

```
sparseMatDateRegion <- purrr::map_df(aMatDateRegion, Matrix::summary, .id = "Type" )
head(sparseMatDateRegion)
```

```
## 517 x 2325 sparse Matrix of class "dgCMatrix", with 152515 entries
##           Type i j      x
## 1 city.driving 1 1 100.00
## 2 city.driving 2 1 100.73
## 3 city.driving 3 1 102.86
## 4 city.driving 4 1 102.65
## 5 city.driving 5 1 109.39
## 6 city.driving 6 1 109.62
```

```
# ggplot2::ggplot(sparseMatDateRegion) +
#   ggplot2::geom_tile( ggplot2::aes( x = j, y = i, fill = log10(x)), color = "white") +
#   ggplot2::scale_fill_gradient(low = "white", high = "blue") +
#   ggplot2::xlab("Region") + ggplot2::ylab("Date") +
#   ggplot2::facet_wrap( ~Type, scales = "free", ncol = 2)
```

```
# d3heatmap::d3heatmap( x = aMatDateRegion[["country/region.driving"]], Rowv = FALSE )
```

```
# th <- 0.94
# aNNGraphs <-
#   purrr::map( aMatDateRegion, function(m) {
#     m2 <- cor(as.matrix(m))
#     for( i in 1:nrow(m2) ) {
#       m2[i,i] <- 0
#     }
#     m2 <- as( m2, "dgCMatrix")
#     #m2@x[ m2@x <= th ] <- 0
#     m2@x[ m2@x > th ] <- 1
#     igraph::graph_from_adjacency_matrix(Matrix::drop0(m2), weighted = TRUE, mode = "undirected")
#   })
```

```
# ind <- 3
# ceb <- cluster_edge_betweenness(aNNGraphs[[ind]])
# dendPlot(ceb, mode="hclust", main = names(aNNGraphs)[[ind]])
```

```
# plot(ceb, aNNGraphs[[ind]], vertex.size=1, vertex.label=NA, main = names(aNNGraphs)[[ind]])
```

Time Series Analysis

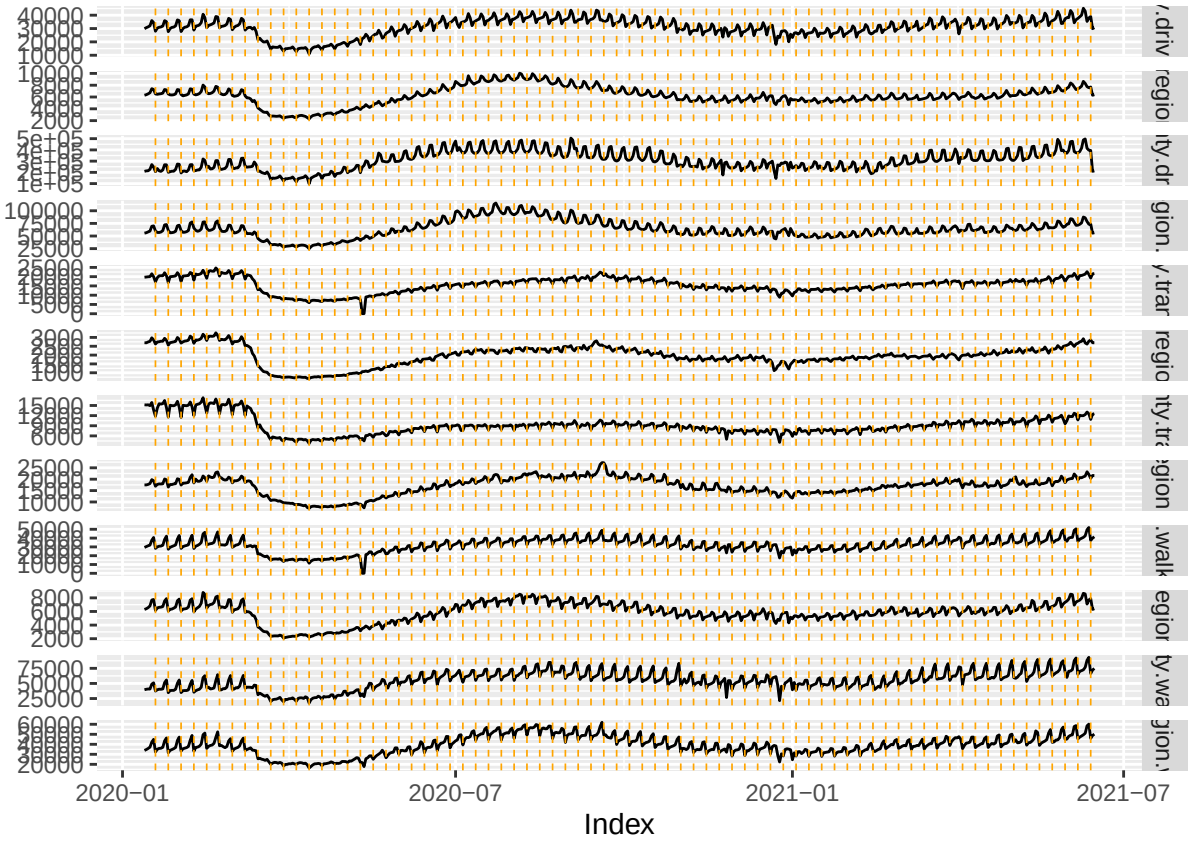
```
aDateStringToDateInfo <- unique(appleMobility_na[, c("Date", "DateInfo")] )
aDateStringToDateInfo <- setNames( aDateStringToDateInfo$DateInfo, aDateStringToDateInfo$Date )
aDateStringToDateInfo <- as.POSIXct(aDateStringToDateInfo, format = "%a %b %d %Y")
aTSDirReqByCountry <-  purrr::map( aMatDateRegion, function(m) rowSums(m) )
```

```
matTS <- do.call( cbind, aTSDirReqByCountry)
```

```
## Warning in (function (..., deparse.level = 1) : number of rows of result is not
## a multiple of vector length (arg 1)
```

```
zooObj <- zoo::zoo( x = matTS, as.POSIXct(rownames(matTS)) )
```

```
autoplot(zooObj) +
  aes(colour = NULL, linetype = NULL) +
    facet_grid(Series ~ ., scales = "free_y") +
  geom_vline( xintercept = aDateStringToDateInfo[weekdays(aDateStringToDateInfo) == "Sunday"], color =
```



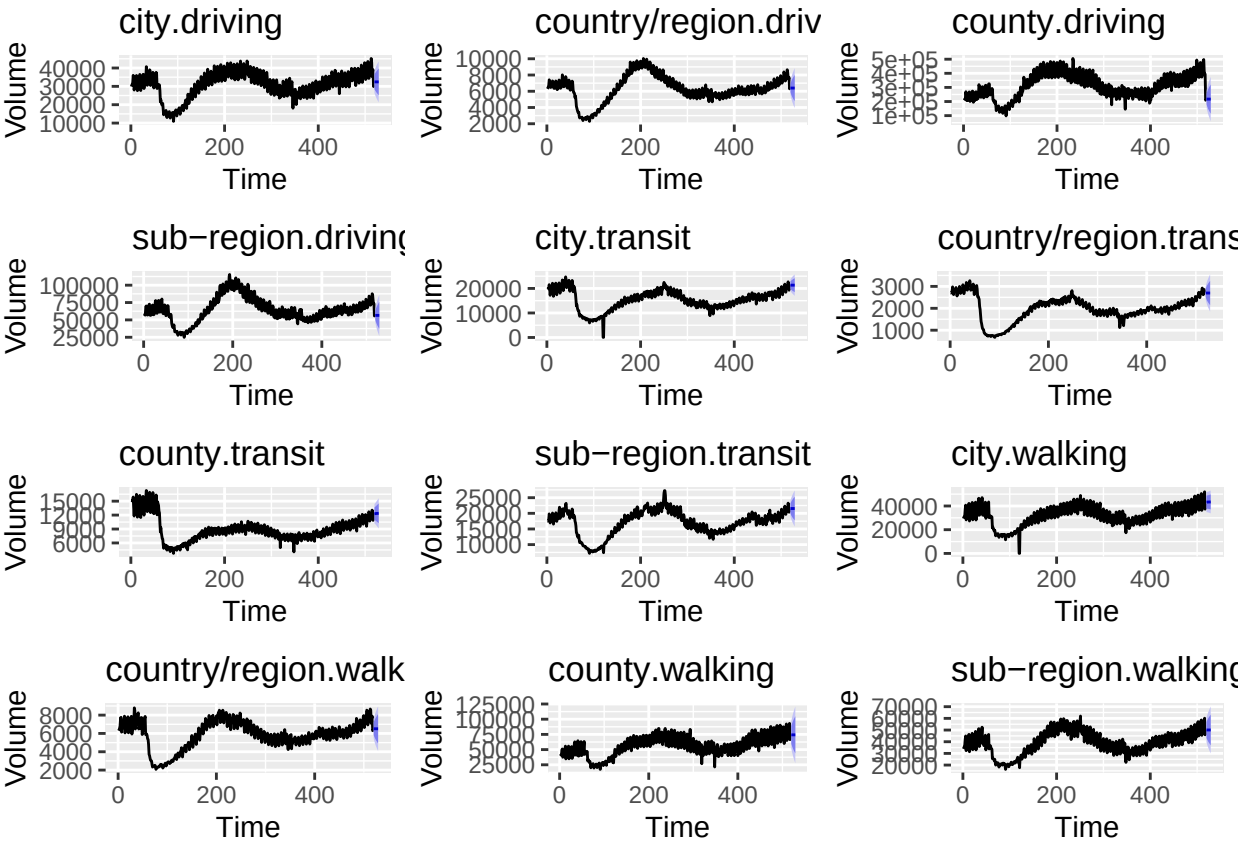
Forecasting

```
aTSModels <- purrr::map( names(zooObj), function(x) { forecast::auto.arima( zoo( x = zooObj[,x], order.
```

```
aTSModels <- purrr::map( names(zooObj), function(x) forecast::forecast( as.matrix(zooObj)[,x] ) )
names(aTSModels) <- names(zooObj)
```

```
lsPlots <- purrr::map( names(aTSModels), function(x) autoplot(aTSModels[[x]]) + ylab("Volume") + ggtitl
names(lsPlots) <- names(aTSModels)
```

```
do.call( gridExtra::grid.arrange, lsPlots )
```



Packages, Repositories, Articles