KJW update_3.10

2022-03-10

Overview

This is a sample analysis on Ofenbach's HSKT stream count over multiple countries. I focus on Stream Count now because Chart Ranking is too conflated with other factors. I am using this song because we have some apriori knoweldge about it. Since it is a dance song, then it peaks in XYZ countries first. I would like to re-create this with two other songs where we know what the pattern is "supposed to be". Do we have another song that we already know peaks in a certain country first? Do we have a song that we know lags far behind other countries? Doesn't have to be a song. An artist, or a genre? For a specific country, or whole territory would do.

Step 1: Data Overivew

I re-shape data of weekly Ofenbach HSKT streams, so each row is a date. Each column is the weekly streams, by country. Sample of the data frame:

```
library(tidyverse)
charts <- read_tsv('/cloud/project/raw/weekly_ghosttown.tsv')
charts_total <- charts %>%
  filter(COUNTRY_CODE %in% c("FR", "US", "GB", "PT")) %>%
  select(COUNTRY_CODE, TOTAL_STREAMS, DATE_KEY)
## Step 1A: reshape
test <- charts_total %>%
  select(TOTAL_STREAMS, COUNTRY_CODE, DATE_KEY) %>%
  group_by_at(vars(-TOTAL_STREAMS)) %>%
  dplyr::mutate(row_id = 1:n()) %>%
  ungroup() %>%
  spread(key = COUNTRY_CODE, value = TOTAL_STREAMS)
test[is.na(test)] = 0
head(test)
```

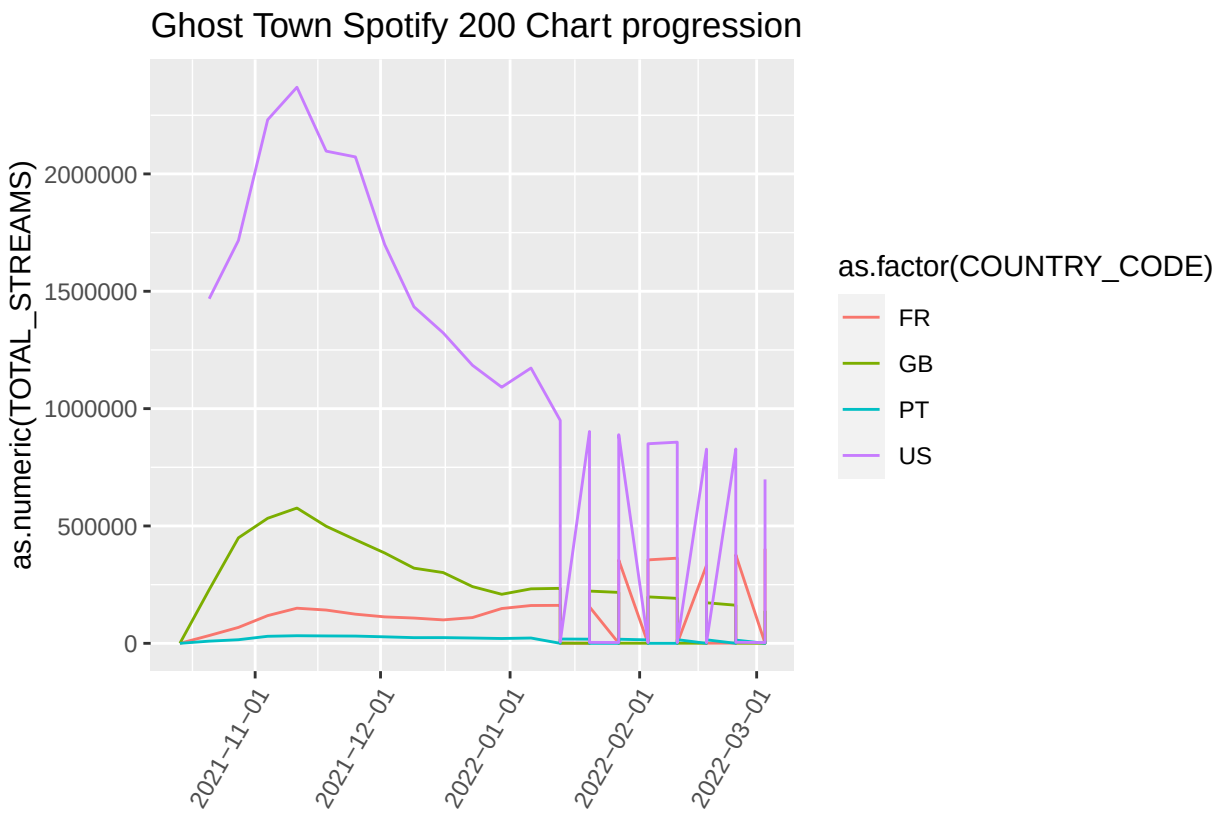
```
## # A tibble: 6 x 6
##   DATE_KEY   row_id     FR     GB    PT      US
##   <date>      <int>  <dbl>  <dbl> <dbl>   <dbl>
## 1 2021-10-14      1     37    129     3       0
## 2 2021-10-21      1  33226 228383  9388 1468174
## 3 2021-10-28      1  67805 449483 15376 1716560
## 4 2021-11-04      1 118073 532732 30000 2230811
## 5 2021-11-11      1 149601 576448 32390 2369275
## 6 2021-11-18      1 141767 498837 31444 2097067
```

Step 2: Pairwise Country Visualizations

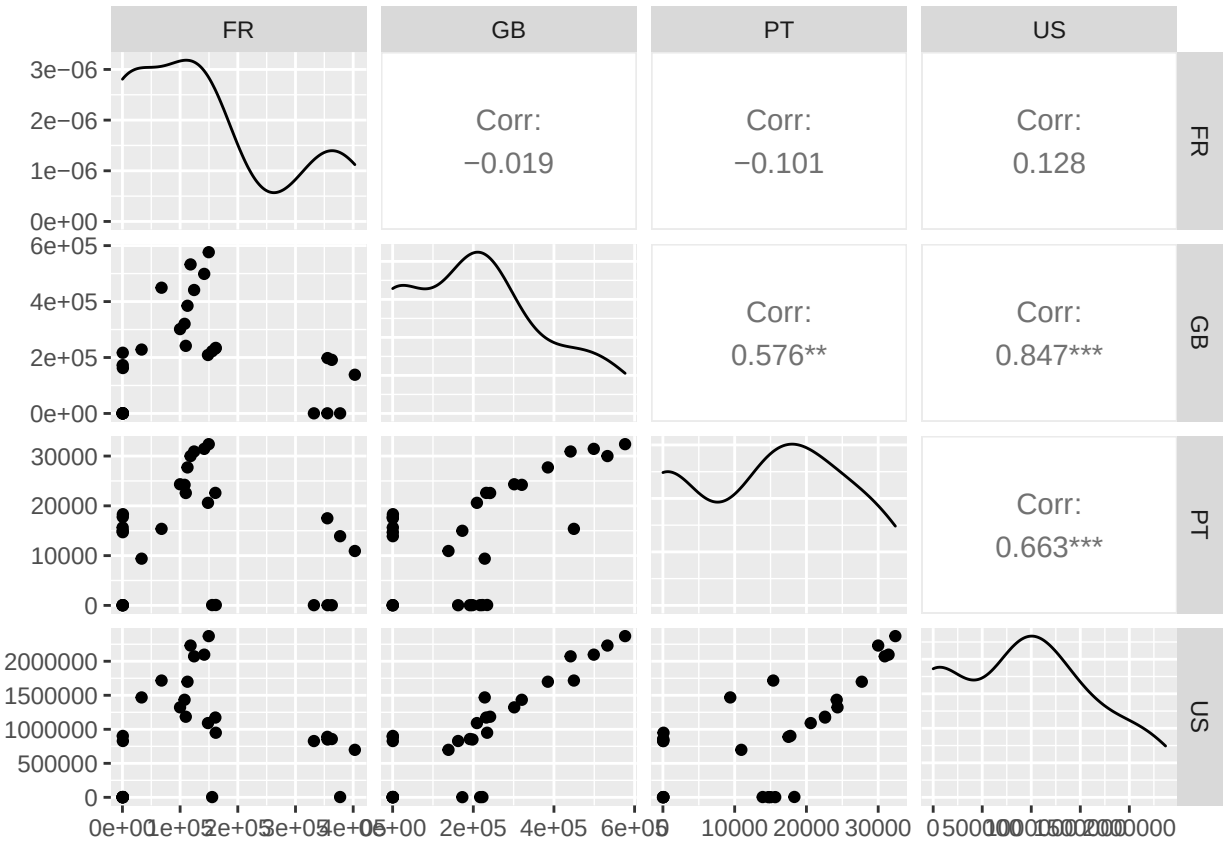
For each pair of countries, visualize the arc of stream count. You can see here that PT peaks after

```
p <- ggplot(charts_total, aes(x=DATE_KEY, y=as.numeric(TOTAL_STREAMS),
                        col = as.factor(COUNTRY_CODE))) +
  geom_line() +
```

```
  xlab("") +
  scale_x_date(date_labels = "%Y-%m-%d") +
  labs(title = "Ghost Town Spotify 200 Chart progression") +
  theme(axis.text.x=element_text(angle=60, hjust=1))
p
```

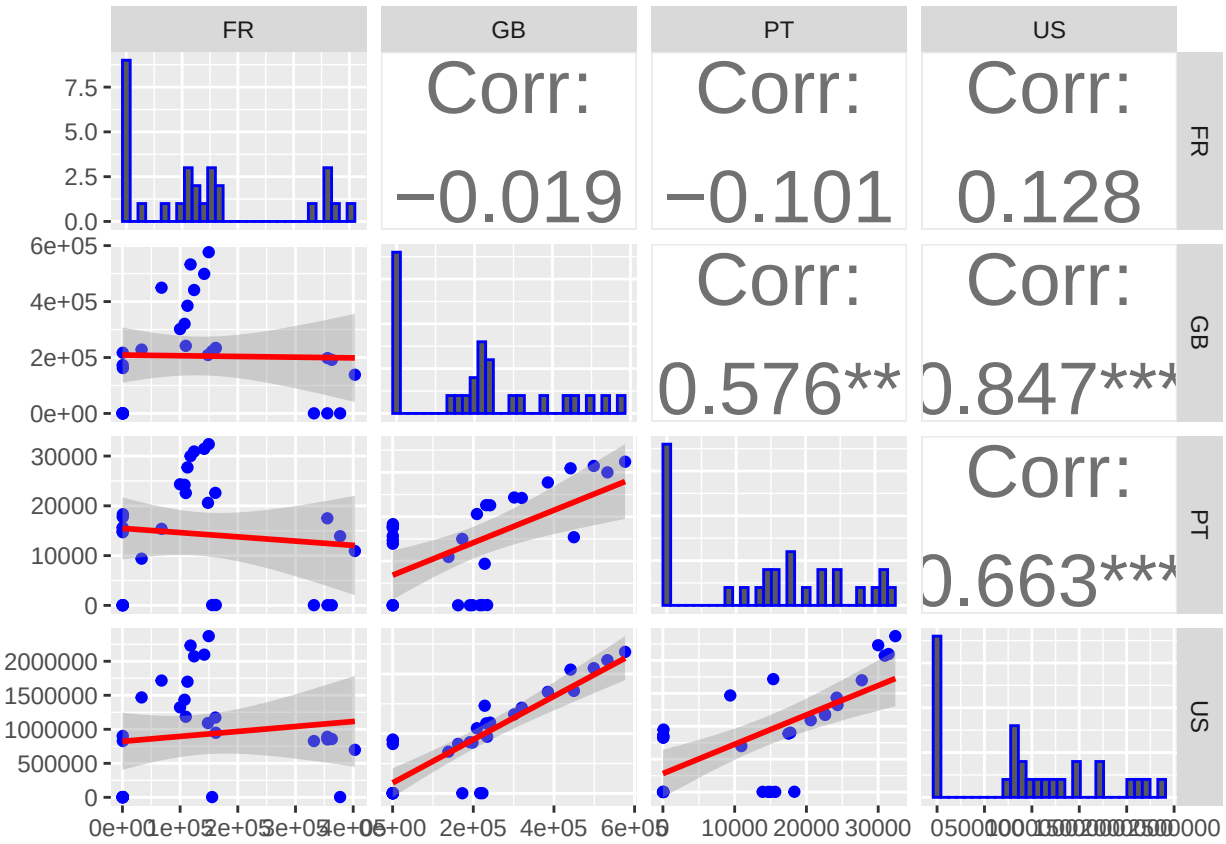



```
## one way
library(GGally)
pairdat<- test[,3:6]
ggpairs(test[,3:6])
```




```
## another way
lowerFn <- function(data, mapping, method = "lm", ...) {
  p <- ggplot(data = pairdat, mapping = mapping) +
    geom_point(colour = "blue") +
    geom_smooth(method = method, color = "red", ...)
  p
}

ggpairs(test[,3:6], lower = list(continuous = wrap(lowerFn, method = "lm")),
  diag = list(continuous = wrap("barDiag", colour = "blue")),
  upper = list(continuous = wrap("cor", size = 10))
)
```



Step 3: Pairwise Country Covariance and Autocorrelation Charts

Covariance/Correlation of the Stream

For one song, we have the vector of stream # for country A and country B. Covariance and correlation is the measure of dependence between the variances

$$Cov[X, y] = \frac{\sum(X_i - \bar{X})(Y_j - \bar{Y})}{n - 1}$$

and Correlation is a standardized measure of Covariance

$$Corr[X, Y] = Cov[X, Y]/\sqrt{Var[X]Var[Y]}$$

Autocorrelation Function

Given by CCF (cross correlation function) and acf (auto-correlation function). The CCF identifies lags of the x-variable that might be useful predictors of $y - t$. The sample CCF is the set of sample correlations between x_{t+h} and y_t for h=0, +-1, +-2, etc. Negative value for h is a correlation between the x variable at a time before t and the y variable at time t. H=-2, then the CCF gives the correlaion between X_{t-2} and y_t.

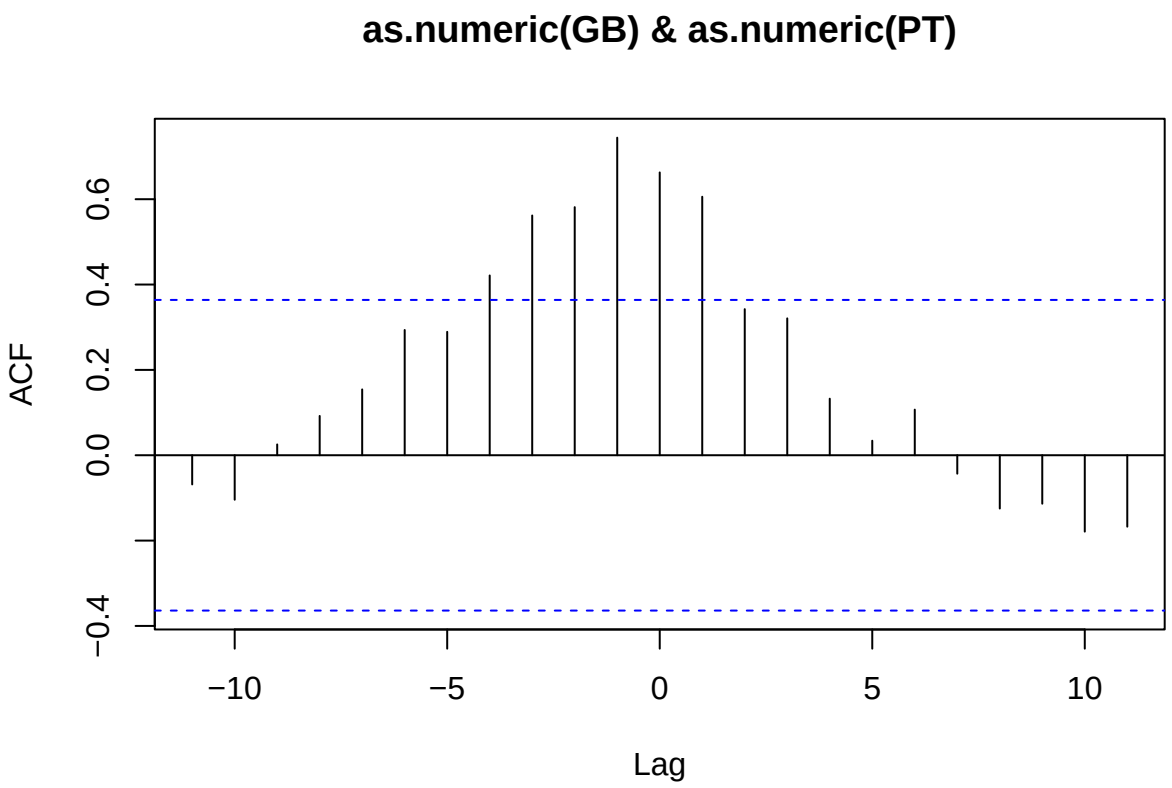
- When one or more x_{t+h}, with h negative, are predictors of y_t, means that x leads y
- When one or more x_{t+h}, with h positive, are predictors of y_t, then xl ags y.

$$CCF(X_t, T_t n) ACT(Y_t, T_t n)$$

read this: https://online.stat.psu.edu/stat510/lesson/8/8.2

For GB and PT, the most dominant cross correlations occur at h=-5 to 5. The maximum correlations in this region are positive, indicating that an above average value of GB streams is likely to lead to an above average value of US streams about 1-2 weeks later.

```
## is GB a potential predictor of PT, positibe correlations
GB = ts(test[6])
PT = ts(test[5])
ccf(as.numeric(GB), as.numeric(PT))
ccfvalues = ccf(as.numeric(GB), as.numeric(PT))
```

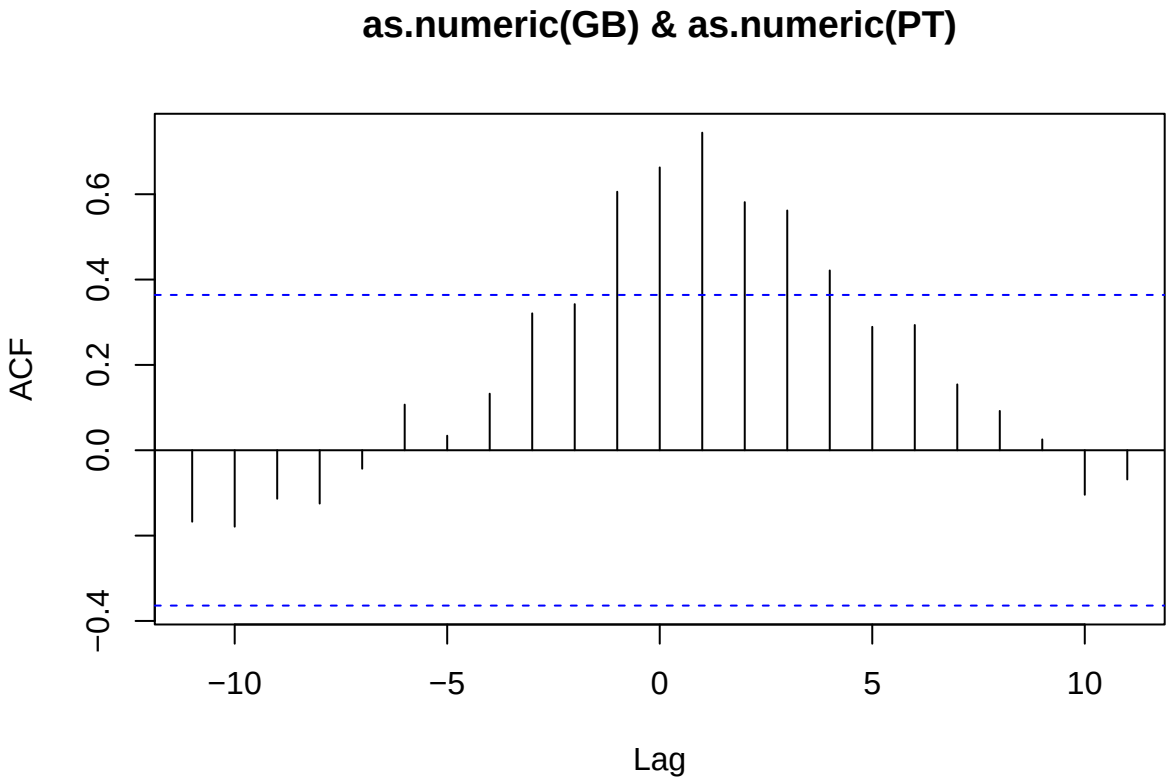



```
ccfvalues
```

```
##
## Autocorrelations of series 'X', by lag
##
##    -11    -10     -9     -8     -7     -6     -5     -4     -3     -2     -1
## -0.068 -0.104  0.025  0.092  0.154  0.294  0.289  0.421  0.562  0.581  0.744
##      0      1      2      3      4      5      6      7      8      9     10
##  0.663  0.606  0.342  0.321  0.133  0.034  0.107 -0.043 -0.125 -0.114 -0.179
##     11
## -0.167
```

If you switch, then does PT predict GB, at later lags, but not before, X lags Y.

```
### is
GB = ts(test[5])
PT = ts(test[6])
ccf(as.numeric(GB), as.numeric(PT))
ccfvalues = ccf(as.numeric(GB), as.numeric(PT))
```




```
ccfvalues
```

```
##
## Autocorrelations of series 'X', by lag
##
##    -11    -10     -9     -8     -7     -6     -5     -4     -3     -2     -1
## -0.167 -0.179 -0.114 -0.125 -0.043  0.107  0.034  0.133  0.321  0.342  0.606
##      0      1      2      3      4      5      6      7      8      9     10
##  0.663  0.744  0.581  0.562  0.421  0.289  0.294  0.154  0.092  0.025 -0.104
##     11
## -0.068
```

Covariance