# update_3.10

2022-03-10

## Data Overivew

The data is DV of stream counts for Ofenbach HSKT

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
charts <- read_tsv('/cloud/project/raw/weekly_ghosttown.tsv')
```

```
## Rows: 4477 Columns: 6
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr  (4): PRODUCT_TITLE, MAJOR_GENRE_DESC, CUSTOMER_NAME, COUNTRY_CODE
## dbl  (1): TOTAL_STREAMS
## date (1): DATE_KEY
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
table(charts$COUNTRY_CODE)
```

```
##
## AD AE AG AL AM AO AR AT AU AZ BA BB BD BE BF BG BH BI BJ BN BO BR BS BT BW BY
## 20 28 21 25 22 22 28 30 30 21 24 21 27 29 20 26 22 21 20 28 24 28 23 28 26 26
## BZ CA CD CG CH CI CL CM CO CR CV CW CY CZ DE DJ DK DM DO DZ EC EE EG ES FI FJ
## 21 28 16 15 29 22 28 20 28 26 21 21 27 29 30 19 29 20 28 28 28 29 28 30 30 23
## FM FR GA GB GD GE GH GM GN GQ GR GT GW GY HK HN HR HT HU ID IE IL IN IQ IS IT
## 19 29 20 29 20 25 28 19 20 19 29 26 16 20 30 24 28 22 29 29 29 29 30 21 28 30
## JM JO JP KE KG KH KI KM KN KR KW KZ LA LB LC LI LK LR LS LT LU LV LY MA MC MD
## 25 29 29 28 21 28 13 18 20 28 25 28 22 26 20 20 28 21 20 26 29 28 18 27 20 23
## ME MG MH MK ML MN MO MR MT MU MV MW MX MY MZ NE NG NI NL NO NP NR NZ OM PA PE
## 22 21 18 24 20 23 25 20 27 25 27 22 28 30 24 18 28 21 30 30 26 14 30 22 27 28
## PG PH PK PL PS PT PW PY QA RO RS RU RW SA SB SC SE SG SI SK SL SM SN SR ST SV
## 23 30 28 30 22 29 20 25 28 29 27 30 21 29 23 21 30 29 28 29 22 19 21 20  6 25
## SZ TD TG TH TJ TL TN TO TR TT TV TW TZ UA UG US UY UZ VC VE VN VU WS XK ZA ZM
## 23 17 21 30 15 21 28 15 28 27  2 29 25 27 24 28 26 22 21 19 29 19 20 21 30 26
## ZW
## 26
```

```
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
```

## Pairwise Country Visualizations

### 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.

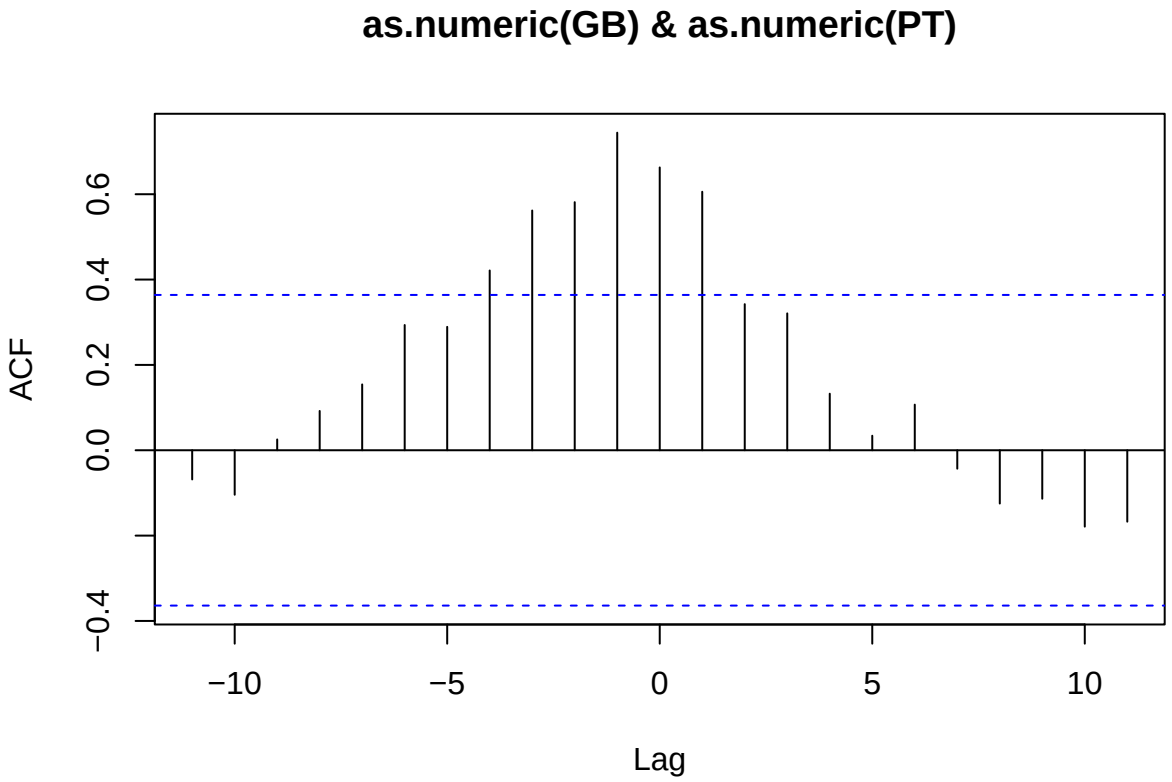
- When one or more $x_{t+h}$, with h negative, are predictors of y_t, means that x leads y
- When one or more $x_{t+h}$, with h positive, are predictors of y_t, then xl ags y.

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

## 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))
```

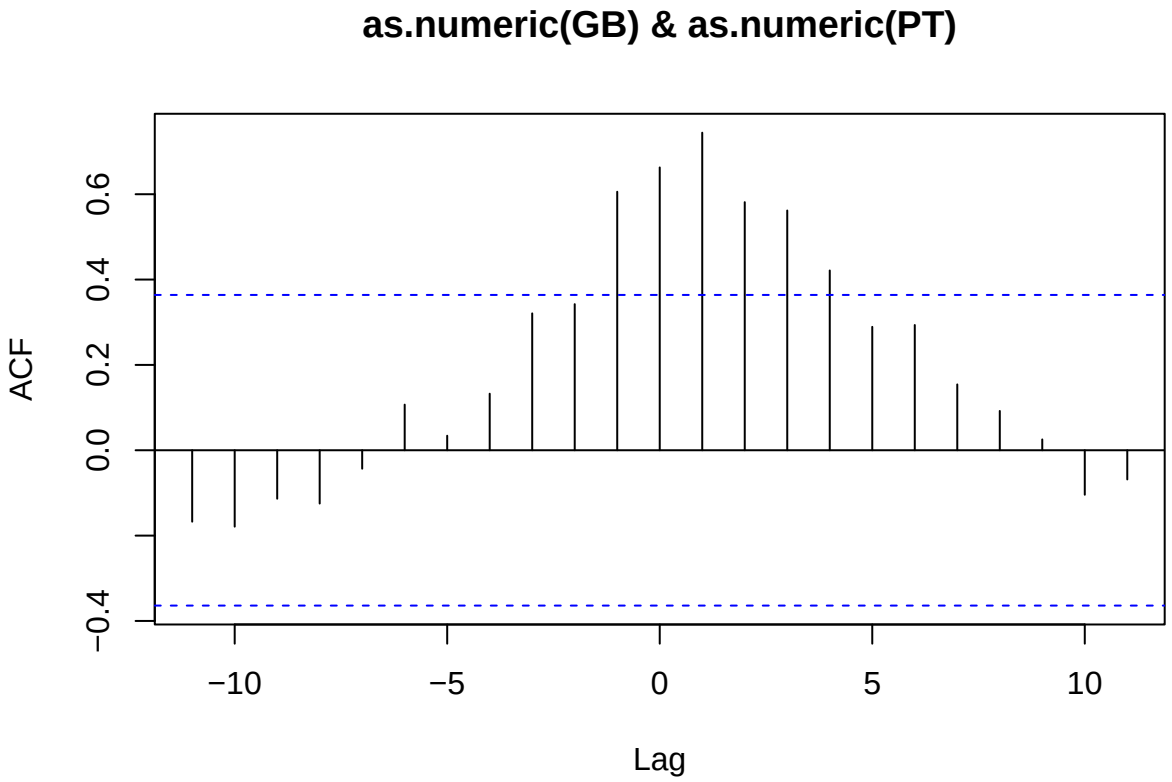



```
ccfvalues
```

```
##
## 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