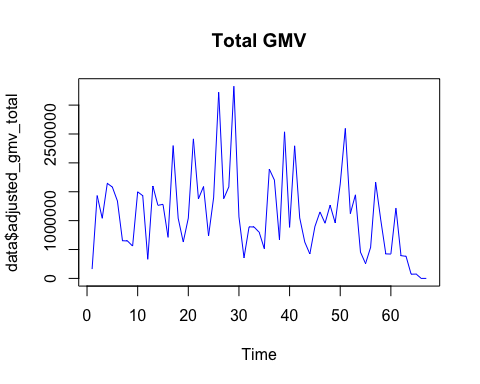
VAR model and IRF Analysis

## Bernardo Carvalho

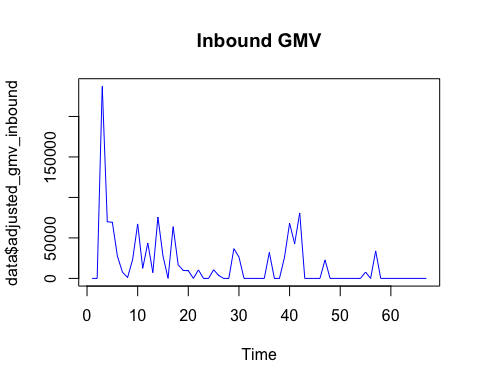
### 07/04/2024

The following code implements the VAR model discussed in the final submission document.

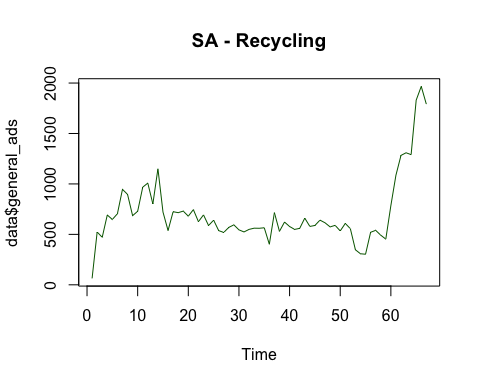
# Rename columns  
data$general\_ads <- data$General  
data$brand\_ads <- data$Brand  
data$display\_ads <- data$Display  
  
ts.plot(data$adjusted\_gmv\_total, col="blue", main="Total GMV")



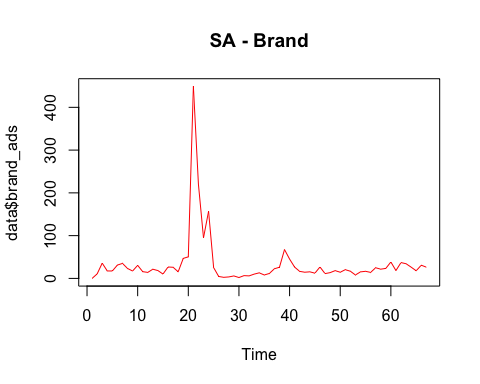
ts.plot(data$adjusted\_gmv\_inbound, col="blue", main="Inbound GMV")



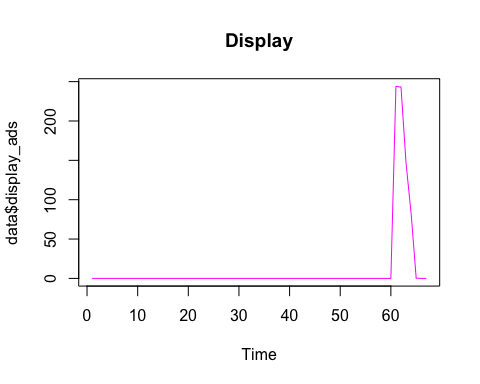
ts.plot(data$general\_ads, col="darkgreen", main="SA - Recycling")



ts.plot(data$brand\_ads, col="red", main="SA - Brand")



ts.plot(data$display\_ads, col="magenta", main="Display")



### Analysis Steps

* Perform unit root tests to check for non-stationary variables and take differences of the variables that are evolving.
* Estimate a vector autoregressive (VAR) model and interpret the coefficients.
* Estimate impulse response functions (IRFs) of sales to spending varialbes

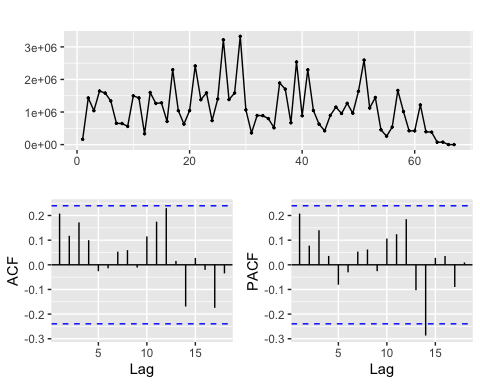
**Log Transformation**

Given the high level of turbulence we observe through time-series plots, we need to take log of each variables to out the series as a first step.

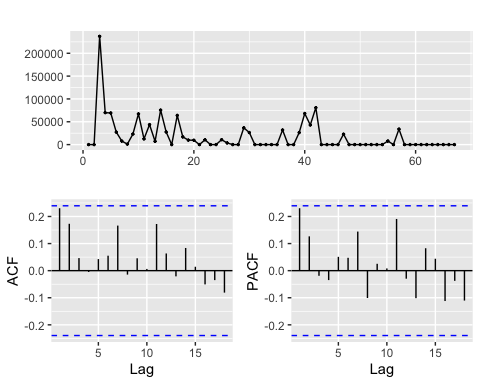
data$log\_adjusted\_gmv\_total <- log(data$adjusted\_gmv\_total+1)  
data$log\_adjusted\_gmv\_inbound <- log(data$adjusted\_gmv\_inbound+1)  
data$log\_general\_ads <- log(data$general\_ads+1)  
data$log\_brand\_ads <- log(data$brand\_ads+1)  
data$log\_display\_ads <- log(data$display\_ads+1)

**Plot the data to observe trend**

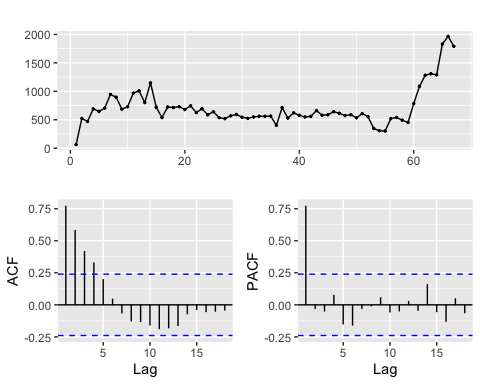
Total\_GMV <- ts(data$adjusted\_gmv\_total, frequency = 1, start = c(1, 1))  
  
#trend plot and ACF and PACF.  
ggtsdisplay(Total\_GMV)



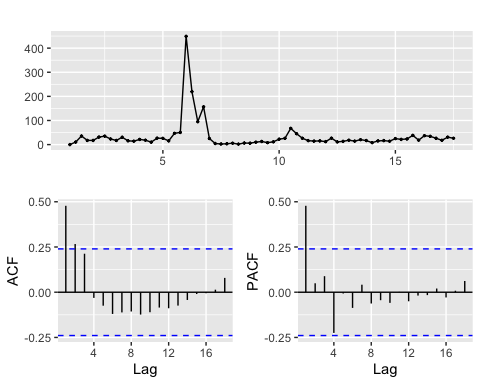
Inbound\_GMV <- ts(data$adjusted\_gmv\_inbound, frequency = 1, start = c(1, 1))  
  
#trend plot and ACF and PACF.  
ggtsdisplay(Inbound\_GMV)



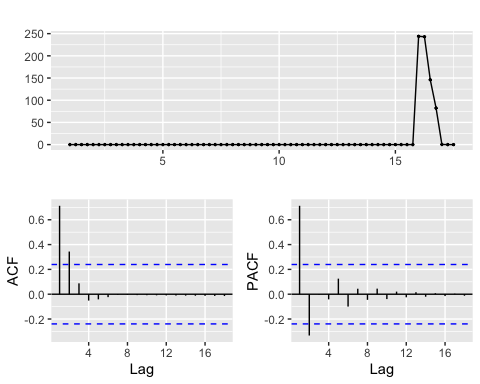
General\_Ads <- ts(data$general\_ads, frequency = 1, start = c(1, 1))  
  
#trend plot and ACF and PACF.  
ggtsdisplay(General\_Ads)



Brand\_Ads <- ts(data$brand\_ads, frequency = 4, start = c(1, 1))  
  
#trend plot and ACF and PACF.  
ggtsdisplay(Brand\_Ads)



Display\_Ads <- ts(data$display\_ads, frequency = 4, start = c(1, 1))  
  
#trend plot and ACF and PACF.  
ggtsdisplay(Display\_Ads)



Based on the adf plots, it seems that only the General ads show non-stationary trends. We test all of these using the adf test:

LgmvTotal <- ts(data$log\_adjusted\_gmv\_total, frequency = 1, start = c(1, 1))  
LgeneralAds <- ts(data$log\_general\_ads, frequency = 1, start = c(1, 1))  
LbrandAds <- ts(data$log\_brand\_ads, frequency = 1, start = c(1, 1))  
LdisplayAds <- ts(data$log\_display\_ads, frequency = 1, start = c(1, 1))  
  
  
adf.test(LgmvTotal)

## Warning in adf.test(LgmvTotal): p-value greater than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: LgmvTotal  
## Dickey-Fuller = 2.0515, Lag order = 4, p-value = 0.99  
## alternative hypothesis: stationary

adf.test(LgeneralAds)

##   
## Augmented Dickey-Fuller Test  
##   
## data: LgeneralAds  
## Dickey-Fuller = -0.48055, Lag order = 4, p-value = 0.98  
## alternative hypothesis: stationary

adf.test(LbrandAds)

##   
## Augmented Dickey-Fuller Test  
##   
## data: LbrandAds  
## Dickey-Fuller = -3.8297, Lag order = 4, p-value = 0.02268  
## alternative hypothesis: stationary

adf.test(LdisplayAds)

##   
## Augmented Dickey-Fuller Test  
##   
## data: LdisplayAds  
## Dickey-Fuller = -1.8237, Lag order = 4, p-value = 0.6466  
## alternative hypothesis: stationary

The test results indicate that only the Brand search is stationary. Therefore, we will need to perform difference transformations to the other time series

DgmvTotal <-diff(data$log\_adjusted\_gmv\_total, differences = 1)  
DgeneralAds <-diff(data$log\_general\_ads, differences = 1)  
DdisplayAds <-diff(data$log\_display\_ads, differences = 1)  
  
adf.test(DgmvTotal)

## Warning in adf.test(DgmvTotal): p-value greater than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: DgmvTotal  
## Dickey-Fuller = 0.25344, Lag order = 4, p-value = 0.99  
## alternative hypothesis: stationary

adf.test(DgeneralAds)

##   
## Augmented Dickey-Fuller Test  
##   
## data: DgeneralAds  
## Dickey-Fuller = -3.7555, Lag order = 4, p-value = 0.027  
## alternative hypothesis: stationary

adf.test(DdisplayAds)

## Warning in adf.test(DdisplayAds): p-value smaller than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: DdisplayAds  
## Dickey-Fuller = -5.319, Lag order = 4, p-value = 0.01  
## alternative hypothesis: stationary

For Total GMV and ads, 1 period difference was not enough. Therefore, we repeat the procedure with 2 periods now

D2gmvTotal <-diff(data$log\_adjusted\_gmv\_total, differences = 2)  
  
adf.test(D2gmvTotal)

##   
## Augmented Dickey-Fuller Test  
##   
## data: D2gmvTotal  
## Dickey-Fuller = -4.032, Lag order = 3, p-value = 0.01386  
## alternative hypothesis: stationary

Stationary tests suggest that now the three variables (with adwords spending first-differenced) are all stationary. We can proceed and construct our VAR model now.

### Construct a VAR model

Let’s start with a VAR model for the Total GMV

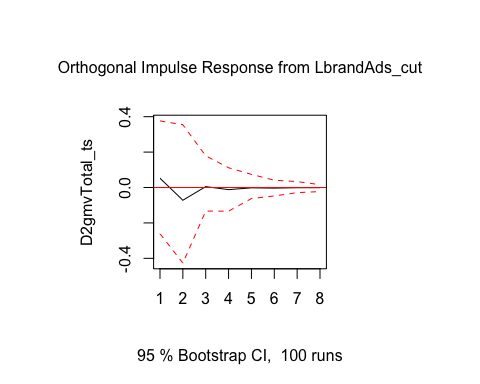
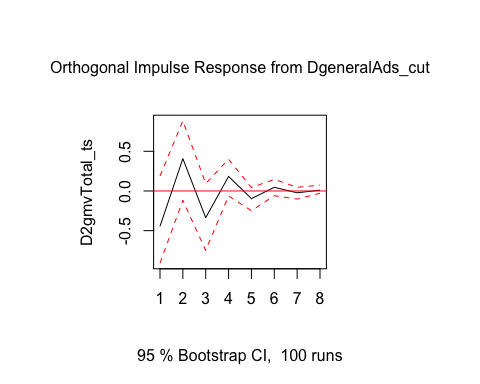
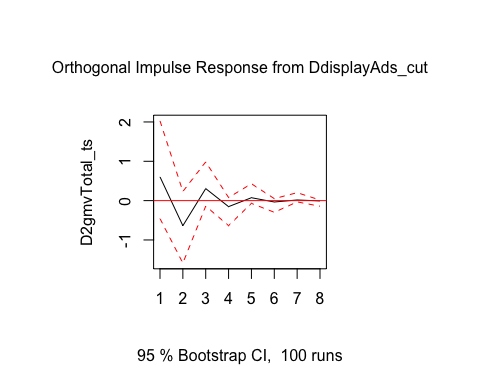
DdisplayAds\_cut <- window(DdisplayAds,start=2)  
DgeneralAds\_cut <- window(DgeneralAds, start=2)   
LbrandAds\_cut <- window(LbrandAds, start=3)  
D2gmvTotal\_ts <- ts(D2gmvTotal, frequency=1)  
  
#Build the first dataset for VAR model  
data.ts.d1 <- cbind(DdisplayAds\_cut, DgeneralAds\_cut, LbrandAds\_cut,   
 D2gmvTotal\_ts)  
  
# Remove NA's  
data.ts.d1 <- replace(data.ts.d1, is.na(data.ts.d1), 0)   
  
var\_total <- VAR(data.ts.d1, ic="AIC", lag.max=1, type="const")  
  
lmp <- var\_total$varresult  
  
#stargazer( lmp$DdisplayAds\_cut, lmp$DgeneralAds\_cut,  
# lmp$LbrandAds\_cut, lmp$D2gmvTotal\_ts,   
 # column.labels = c('Display', 'General', 'Brand', 'GMV'),   
# type = "text", dep.var.labels.include = FALSE )  
  
stargazer(lmp, type='text')

##   
## =================================================================  
## Dependent variable:   
## -----------------------------------  
## y   
## (1) (2) (3) (4)   
## -----------------------------------------------------------------  
## DdisplayAds\_cut.l1 0.062 0.015 -0.070 -0.391   
## (0.128) (0.032) (0.101) (0.286)   
##   
## DgeneralAds\_cut.l1 1.037\*\* -0.182 0.338 0.884   
## (0.514) (0.127) (0.409) (1.153)   
##   
## LbrandAds\_cut.l1 -0.043 -0.024 0.637\*\*\* -0.068   
## (0.107) (0.026) (0.085) (0.240)   
##   
## D2gmvTotal\_ts.l1 0.016 -0.006 0.050 -0.518\*\*\*  
## (0.048) (0.012) (0.038) (0.107)   
##   
## const 0.107 0.092 1.121\*\*\* 0.172   
## (0.334) (0.083) (0.265) (0.748)   
##   
## -----------------------------------------------------------------  
## Observations 66 66 66 66   
## R2 0.069 0.050 0.491 0.352   
## Adjusted R2 0.008 -0.013 0.457 0.309   
## Residual Std. Error (df = 61) 0.858 0.212 0.682 1.922   
## F Statistic (df = 4; 61) 1.133 0.795 14.691\*\*\* 8.275\*\*\*   
## =================================================================  
## Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

### IRF Analysis[[1]](#footnote-53)

First, let’s have a look at the graphs. IRF plots help you see visually when the peak effects occur. On the plots, the black, solid line refers to IRF coefficients, while the red, dashed lines refer to lower and upper bound of the IRF coefficient’s confidence interval.

irfs <- irf(var\_total, impulse = c('DdisplayAds\_cut', 'DgeneralAds\_cut',   
 'LbrandAds\_cut'),   
 response = 'D2gmvTotal\_ts',   
 runs = 100, n.ahead = 7 , ortho = TRUE, ci=0.95)  
plot(irfs)



**Immediate and Long-Term Effects**

In order to compute the immediate and long-term effects, the following steps should be taken.

* Evaluate the significance of each IRF coefficient. If the t-statistics of the IRF coefficient is greater than 1 (t-stat>1)[[2]](#footnote-63)

Then we treat it as signficant and keep the value of that coefficient; otherwise, we treat the coefficient as zero.

To calculate the t-statistics, we need to:

* Derive the standard error of each coefficient from its confidence interval, since
* Calculate the t-statistics using
  + .

Based on the above computations, the first period impact is called the immediate effect while the cumulative effect over 8 periods is called the long-run effect.

Now we make a table to summarize IRF coefficients and their confidence intervals. You will see in the output that *response* means the reponse value at a particular period (there are 8 periods in total), *lower* and *upper* refer to the lower and upper bound of the corresponding confidence intervals, respectively.

#Make a table to summarize IRF coefficients and their confidence intervals  
  
irf.table.ci <- round(data.frame(period = seq(1, 8),   
 response.display\_ads = irfs$irf$DdisplayAds\_cut,   
 display\_ads.lower = irfs$Lower$DdisplayAds\_cut,   
 display\_ads.upper = irfs$Upper$DdisplayAds\_cut,  
 response.general\_ads = irfs$irf$DgeneralAds\_cut,   
 general\_ads.lower = irfs$Lower$DgeneralAds\_cut,   
 general\_ads.upper = irfs$Upper$DgeneralAds\_cut,   
 response.brand\_ads = irfs$irf$LbrandAds\_cut,   
 brand\_ads.lower = irfs$Lower$LbrandAds\_cut,   
 brand\_ads.upper = irfs$Upper$LbrandAds\_cut)  
 ,4)  
colnames(irf.table.ci) <- c(  
 'Period', 'Display', 'Display Lower', 'Display Upper',  
 'General', 'General Lower', 'General Upper',   
 'Brand','Brand Lower', 'Brand Upper')  
  
knitr::kable(irf.table.ci)

| Period | Display | Display Lower | Display Upper | General | General Lower | General Upper | Brand | Brand Lower | Brand Upper |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0.6033 | -0.4573 | 2.0271 | -0.4457 | -0.9090 | 0.1885 | 0.0522 | -0.2611 | 0.3763 |
| 2 | -0.6377 | -1.5886 | 0.2320 | 0.4070 | -0.1182 | 0.8830 | -0.0718 | -0.4270 | 0.3541 |
| 3 | 0.3059 | -0.1392 | 0.9845 | -0.3367 | -0.7493 | 0.0937 | 0.0050 | -0.1331 | 0.1789 |
| 4 | -0.1529 | -0.6356 | 0.0784 | 0.1833 | -0.0627 | 0.3999 | -0.0120 | -0.1336 | 0.1116 |
| 5 | 0.0740 | -0.0695 | 0.4292 | -0.0966 | -0.2491 | 0.0439 | -0.0019 | -0.0624 | 0.0742 |
| 6 | -0.0349 | -0.2962 | 0.0509 | 0.0461 | -0.0593 | 0.1455 | -0.0034 | -0.0476 | 0.0412 |
| 7 | 0.0167 | -0.0325 | 0.2019 | -0.0230 | -0.1021 | 0.0462 | -0.0012 | -0.0290 | 0.0333 |
| 8 | -0.0077 | -0.1422 | 0.0156 | 0.0104 | -0.0283 | 0.0733 | -0.0012 | -0.0230 | 0.0161 |

Now we apply the t>1 criteria to determine coefficient significance and calculate long-term elasticities of adwords and flyer advertising spending.

#General Adwords  
result\_irf\_general\_adwords<-matrix(nrow = 8, ncol = 1)  
  
for (i in 1:8) {  
 se <- (irfs$Upper$DdisplayAds\_cut[i]-irfs$Lower$DdisplayAds\_cut[i])/(2\*1.96)  
 t\_irf\_general\_adwords<- irfs$irf$DdisplayAds\_cut[i]/se  
   
 if (t\_irf\_general\_adwords>1) {  
 result\_irf\_general\_adwords[i] <- irfs$irf$DdisplayAds\_cut[i]  
 } else {  
 result\_irf\_general\_adwords[i] <-0  
 }  
}  
  
result\_irf\_general\_adwords #print out the results

## [,1]  
## [1,] 0.0000000  
## [2,] 0.0000000  
## [3,] 0.3058828  
## [4,] 0.0000000  
## [5,] 0.0000000  
## [6,] 0.0000000  
## [7,] 0.0000000  
## [8,] 0.0000000

lr\_general\_adwords <- sum(result\_irf\_general\_adwords)  
lr\_general\_adwords

## [1] 0.3058828

#Brand Adwords  
result\_irf\_brand\_adwords<-matrix(nrow = 8, ncol = 1)  
  
for (i in 1:8) {  
 se <- (irfs$Upper$LbrandAds\_cut[i]-irfs$Lower$LbrandAds\_cut[i])/(2\*1.96)  
 t\_irf\_brand\_adwords<- irfs$irf$LbrandAds\_cut[i]/se  
   
 if (t\_irf\_brand\_adwords>1) {  
 result\_irf\_brand\_adwords[i] <- irfs$irf$LbrandAds\_cut[i]  
 } else {  
 result\_irf\_brand\_adwords[i] <-0  
 }  
}  
  
result\_irf\_brand\_adwords #print out the results

## [,1]  
## [1,] 0  
## [2,] 0  
## [3,] 0  
## [4,] 0  
## [5,] 0  
## [6,] 0  
## [7,] 0  
## [8,] 0

lr\_brand\_adwords <- sum(result\_irf\_brand\_adwords)  
lr\_brand\_adwords

## [1] 0

#Brand Adwords  
result\_irf\_brand\_adwords<-matrix(nrow = 8, ncol = 1)  
  
for (i in 1:8) {  
 se <- (irfs$Upper$LbrandAds\_cut[i]-irfs$Lower$LbrandAds\_cut[i])/(2\*1.96)  
 t\_irf\_brand\_adwords<- irfs$irf$LbrandAds\_cut[i]/se  
   
 if (t\_irf\_brand\_adwords>1) {  
 result\_irf\_brand\_adwords[i] <- irfs$irf$LbrandAds\_cut[i]  
 } else {  
 result\_irf\_brand\_adwords[i] <-0  
 }  
}  
  
result\_irf\_brand\_adwords #print out the results

## [,1]  
## [1,] 0  
## [2,] 0  
## [3,] 0  
## [4,] 0  
## [5,] 0  
## [6,] 0  
## [7,] 0  
## [8,] 0

lr\_brand\_adwords <- sum(result\_irf\_brand\_adwords)  
lr\_brand\_adwords

## [1] 0

1. In this session we are using *orthogonalized impulse reaction function* for estimation; specifically, the model requires that a shock occurs only in one variable at a time. Such an assumption is reasonable if the shocks in different variables are independent. Moreover, the method implies certain ordering of the variables of interest. Under certain circumstances, the above-mentioned assumptions and restrictinos are not very reasonable. That’s why one might consider adopting *generalized impulse reaction function (GIRF)*, which will loose these assumptions. However, in R environment, to implement GIRF estimations, we need to estimate a Bayesian VAR model (you may check pacakge “bvartools”) instead, which is beyond the scope of this session. [↑](#footnote-ref-53)
2. Here we follow previous research (e.g., Slotegraaf and Pauwels, 2008) to set the criteria as t>1. You may apply t>2 rule if you would like to evaluate coefficient significanc at 95% significance level. [↑](#footnote-ref-63)