ARIMA model Spanish nett mass cucumber imports into the UK

Required libraries

```
require("RPostgreSQL")
```

```
## Loading required package: RPostgreSQL
```

```
## Loading required package: DBI
```

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.2.1 ──
```

```
## ✓ tibble  1.4.2     ✓ purrr   0.2.4
## ✓ tidyr   0.8.0     ✓ stringr 1.3.0
## ✓ readr   1.1.1     ✓ forcats 0.3.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
library(tseries)
library(forecast)
```

Get the HMRC auxiliary data

```
source("/Users/Warren/S2DS/src/data/get_HMRC_aux_data.R")
list1   <- get_HMRC_aux_data()
comcode <- data.frame(Reduce(rbind, list1[1]))
```

```
port    <- data.frame(Reduce(rbind, list1[2]))
country <- data.frame(Reduce(rbind, list1[3]))
```

Find cucumber comcodes - Thank you Alex for making life nice and easy for us!

```
cc_all_cucumber <- comcode[grep('CUCUMBER',  toupper(comcode$description)),]
```

Get HMRC import data from EU (arrivals df)

```
source("/Users/Warren/S2DS/src/data/get_HMRC_data.R")
HMRC_EU_import_food_data <- get_HMRC_data(arrivals)

## [1] "Medium cuppa?"

(col_names               <-
t(as.data.frame(colnames(HMRC_EU_import_food_data))))

##                                    [,1]          [,2]
## colnames(HMRC_EU_import_food_data) "smk_comcode" "smk_record_type"
##                                    [,3]          [,4]
## colnames(HMRC_EU_import_food_data) "smk_cod_seq" "smk_cod_alpha"
##                                    [,5]            [,6]
## colnames(HMRC_EU_import_food_data) "smk_trade_ind" "smk_coo_seq"
##                                    [,7]
## colnames(HMRC_EU_import_food_data) "smk_coo_alpha"
##                                    [,8]
## colnames(HMRC_EU_import_food_data) "smk_nature_of_transaction"
##                                    [,9]
## colnames(HMRC_EU_import_food_data) "smk_mode_of_transport"
##                                    [,10]
## colnames(HMRC_EU_import_food_data) "smk_period_reference"
##                                    [,11]                [,12]
## colnames(HMRC_EU_import_food_data) "smk_suite_indicator" "smk_sitc"
##                                    [,13]
## colnames(HMRC_EU_import_food_data) "smk_ip_comcode"
##                                    [,14]
## colnames(HMRC_EU_import_food_data) "smk_no_of_consignments"
##                                    [,15]            [,16]
## colnames(HMRC_EU_import_food_data) "smk_stat_value" "smk_nett_mass"
##                                    [,17]
## colnames(HMRC_EU_import_food_data) "smk_supp_unit"
```

Select Spain cucumber import info

```
HMRC_columns  <- col_names[c(1,4,10,14,15,16)]
HMRC_country  <- "ES"
HMRC_comcode  <- "07070005"
model_data    <- HMRC_EU_import_food_data %>%
```

```
                  select(HMRC_columns) %>%
filter(smk_cod_alpha == HMRC_country & smk_comcode ==
HMRC_comcode) %>%
                  select(-smk_cod_alpha, -smk_comcode)
```

Clean data: remove any Na values, 0 values and date values == "0000000"

```
model_data_clean <- model_data %>% na.omit %>%
                    filter(smk_nett_mass!=0 & smk_no_of_consignments != 0 &
smk_stat_value != 0) %>%
filter(smk_period_reference!="0000000") %>%
mutate(abs(smk_nett_mass) & abs(smk_no_of_consignments) &
abs(smk_stat_value)) %>%
                    unique()
```

Convert period column to date format

```
model_data_clean <- model_data_clean %>%
                    mutate(smk_period_reference =
substr(smk_period_reference,                                    4,7)) %>%
                    mutate(smk_period_reference =
paste0(smk_period_reference,"01")) %>%
                    mutate(smk_period_reference =
as.Date(smk_period_reference,
                                                              "%y%m%d"))
```

Get the cos/kg ($/kg)

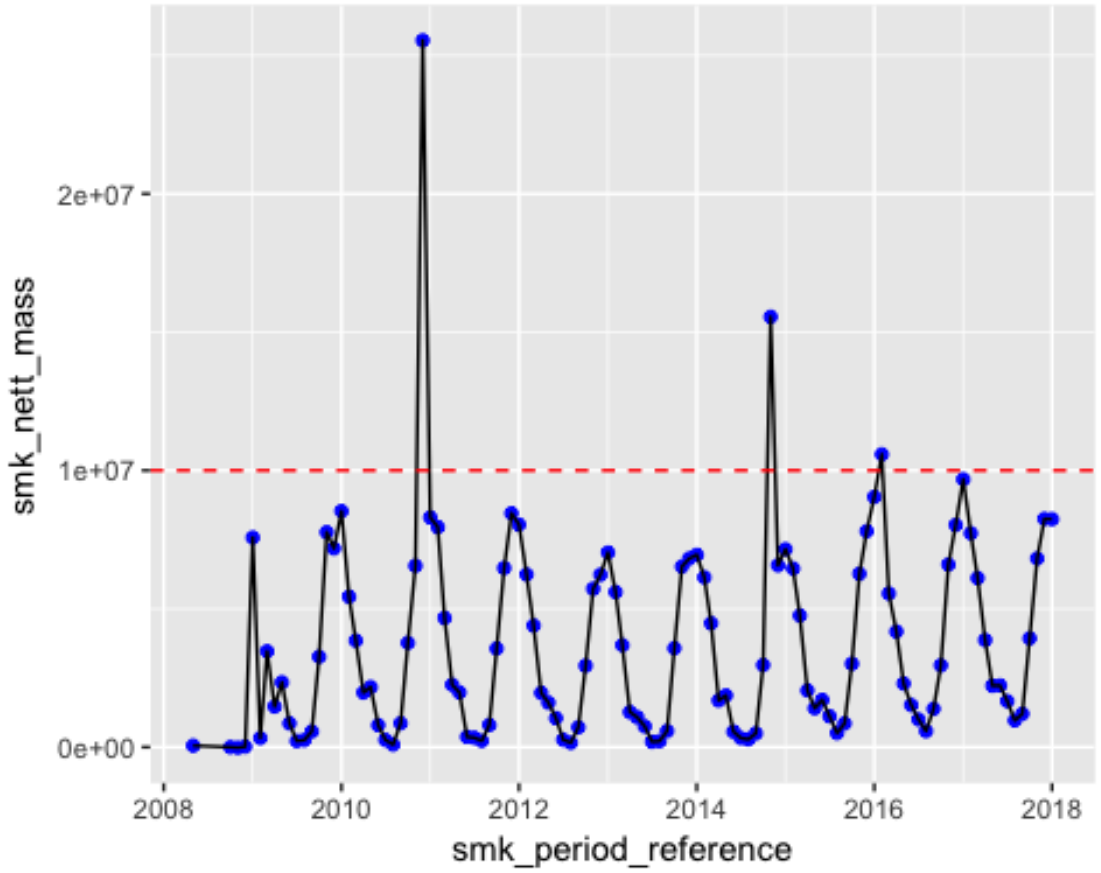
```
model_data_clean <- model_data_clean %>%
                    mutate(cost_per_kg = smk_stat_value / smk_nett_mass)
```

Aggregate data into periods of months

```
model_data_group <- model_data_clean %>% group_by(smk_period_reference) %>%
summarise(smk_nett_mass = sum(smk_nett_mass),
                    smk_no_of_consignments = sum(smk_no_of_consignments),
                    smk_stat_value = sum(smk_stat_value), cost_per_kg =
mean(cost_per_kg))
```

What does the nett mass of the cucumber imported into the UK look like over time?

```
ggplot(model_data_group, aes(smk_period_reference, smk_nett_mass)) +
        geom_point(colour = "blue") +
        geom_line() +
        geom_hline(yintercept=1E7, linetype="dashed", color = "red")
```



Is it possible to model seasonal imports of cucumbers from Spain into the UK based on previous data? For time series data, auto regressive moving average models (ARIMA) can be used to model and forecast. Very good resource for ARMIA model explanation and statistics in general https://onlinecourses.science.psu.edu/stat510/node/67

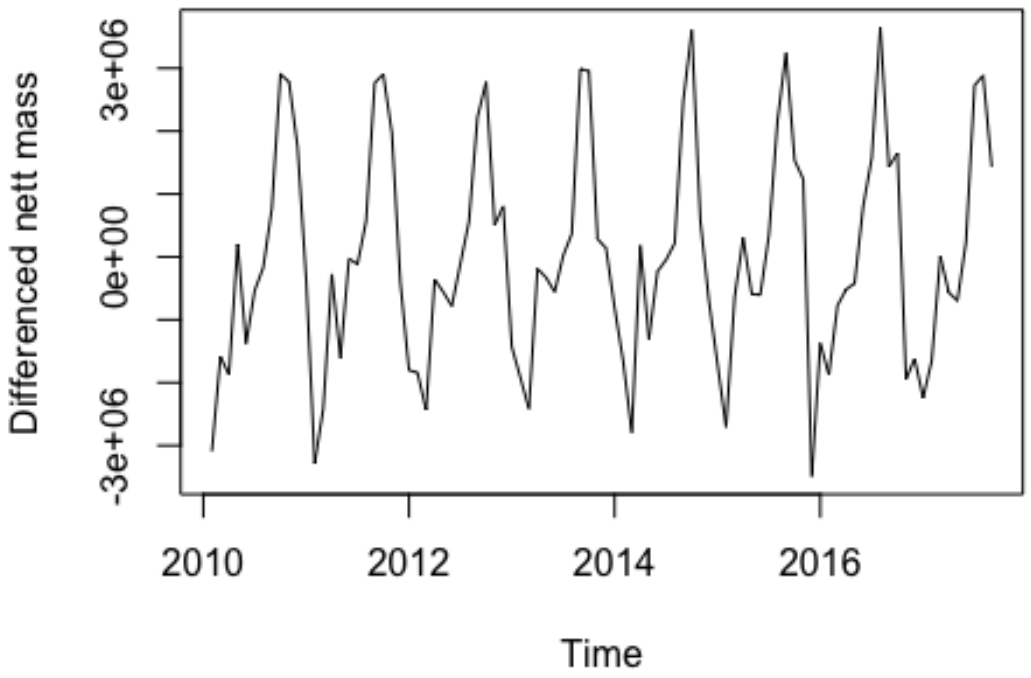
ARIMA model for nett mass imports. Exclude outliers

```
ARIMA_model <-  model_data_group %>% na.omit %>% filter(smk_nett_mass<1E7)
%>%                          mutate(smk_nett_mass=abs(smk_nett_mass)) %>%
filter(smk_period_reference                    > "2009-12-31" &
smk_period_reference <"2017-12-31")

data <- ts(ARIMA_model[, c('smk_nett_mass')], start = c(2012-01-01),
frequency = 12)
```

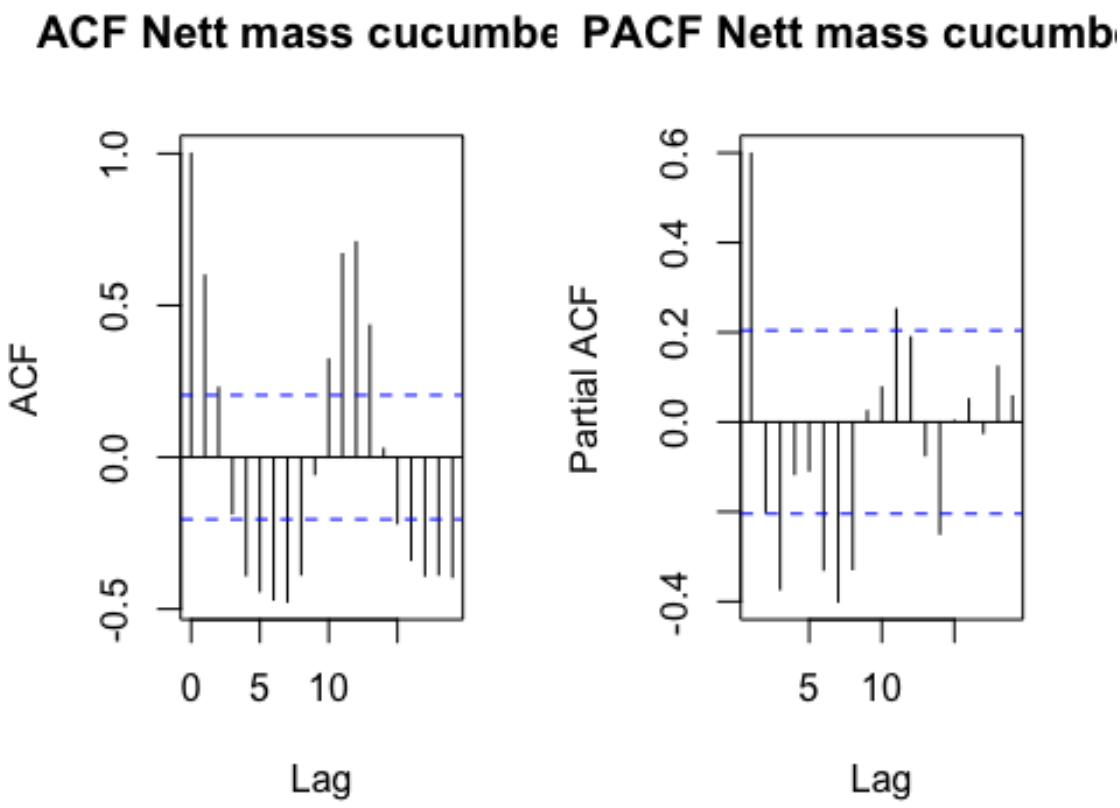
Check if data stationary

```
plot(diff(data),ylab="Differenced nett mass")
```



Create ACF and PACF plots - use to tune parameters in forecast model. Still need to learn more about this and how to tune model using these plots

```
par(mfrow = c(1,2))
acf(ts(diff((data))),main='ACF Nett mass cucumbers')
pacf(ts(diff((data))),main='PACF Nett mass cucumbers')
```



ARIMA model - use auto.fit function from forecast package. Can use arima function and include model order etc and parameter estimates starting points from ACF and PACF plots

```
ARIMAfit <- auto.arima(data)
summary(ARIMAfit)
```

```
## Series: data
## ARIMA(1,0,0)(0,1,0)[12]
##
## Coefficients:
##          ar1
##       0.5837
## s.e.  0.0888
##
## sigma^2 estimated as 8.468e+11:  log likelihood=-1226.96
## AIC=2457.92   AICc=2458.07   BIC=2462.71
##
## Training set error measures:
##                    ME     RMSE    MAE        MPE     MAPE      MASE
## Training set 57098.91 853478.9 584352 0.04806474 32.06352 0.7306809
```

```
##                       ACF1
## Training set 0.005085565
```

Forecast - use model to forecast (h in months,blue line) with 95% confidence (level, grey shade)

```
par(mfrow <- c(1,1))
```

```
## NULL
```

```
pred <- forecast(ARIMAfit, h=24, level =95)
plot(pred)
```