Time Series HW6

106070020

2021年5月10日

姓名：李嘉蓉、何羿樺

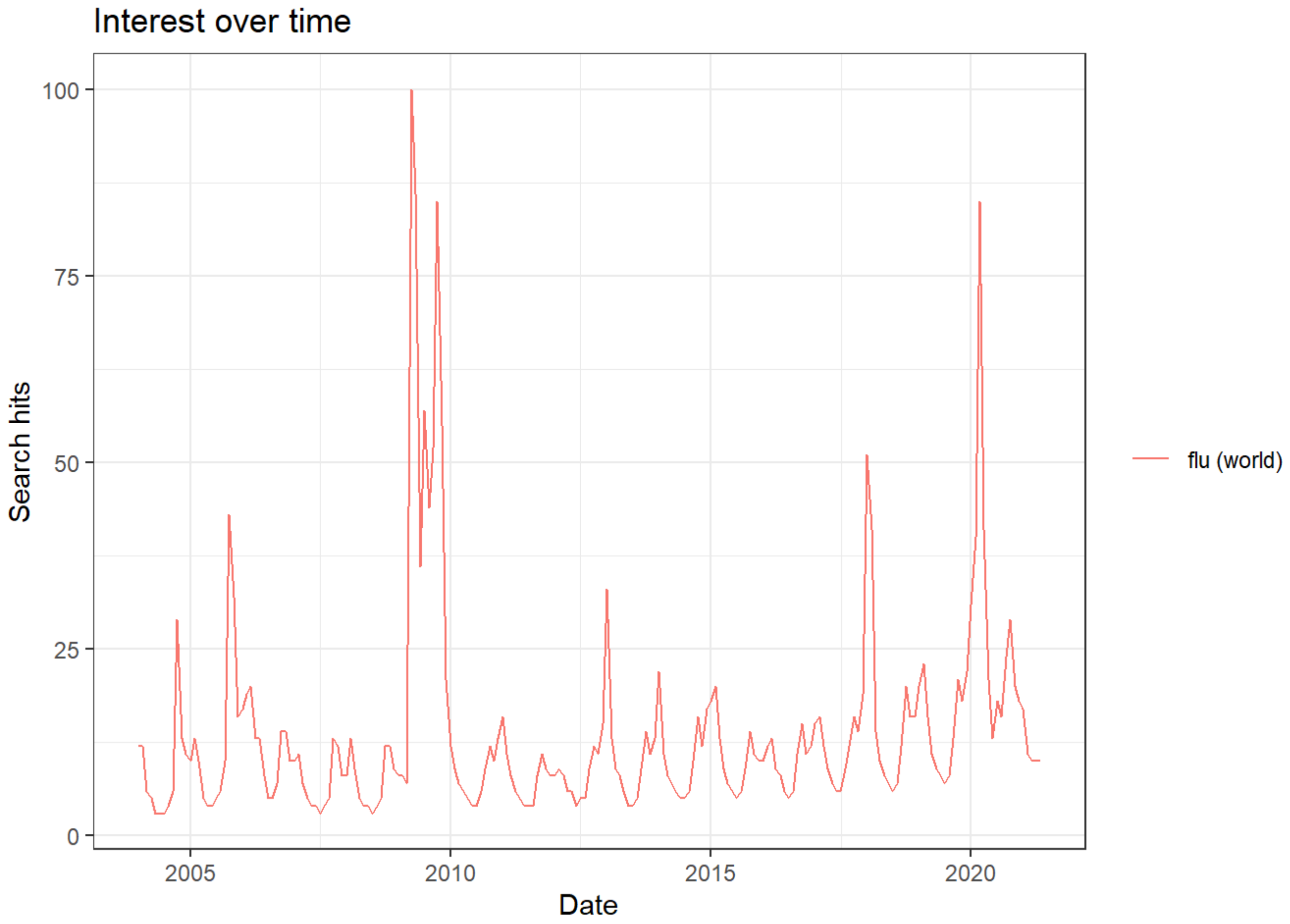
```
library(gtrendsR)
library(astsa)
library(aTSA)
library(forecast)
library(strucchange)
library(ggplot2)
library(ggthemes)
library(forecast)
library(tseries)
library(Metrics)
```

Dataset Flu

Basic information

此為關鍵字為flu之資料集，並針對其作線圖及直方圖。

```
x = gtrends("flu", time="all")
plot(x)
```



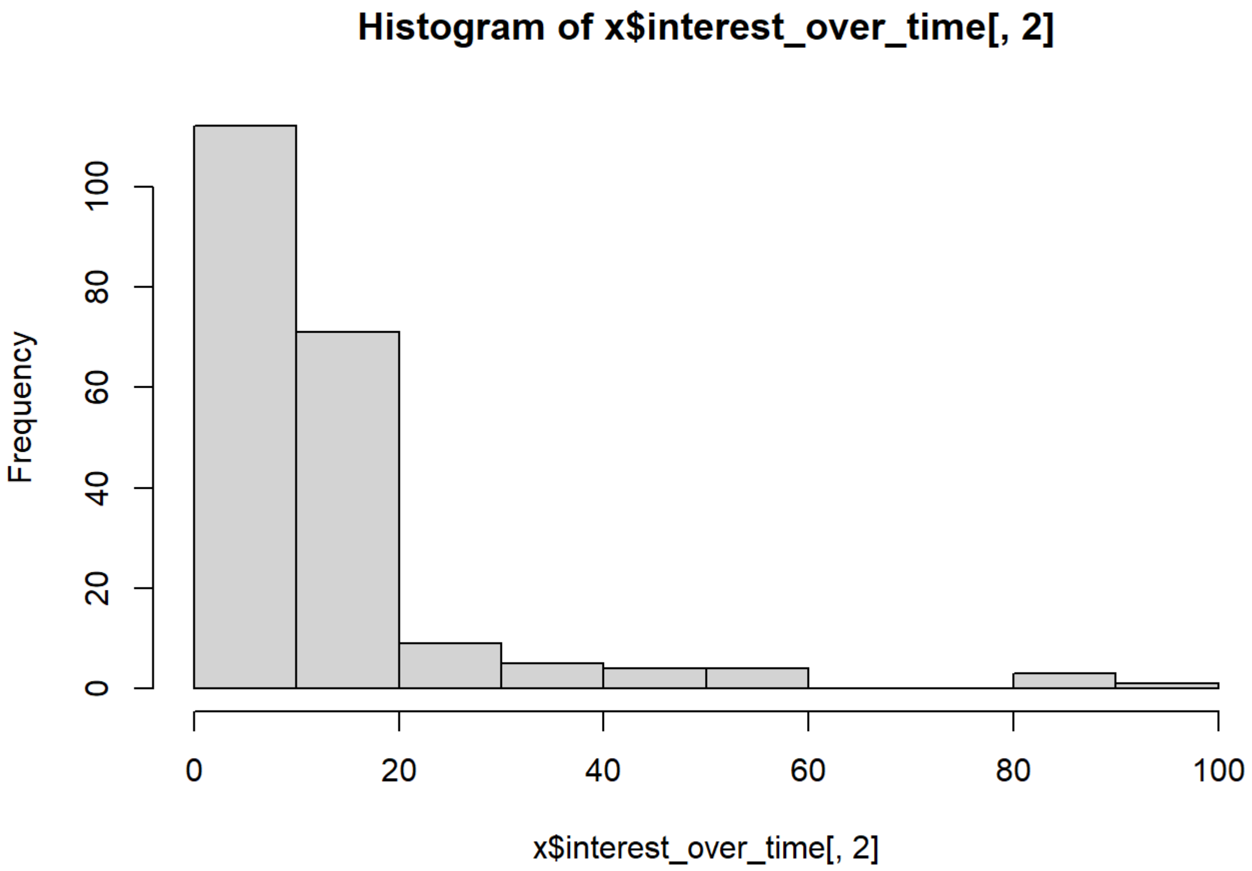
```
names(x)
```

```
## [1] "interest_over_time"   "interest_by_country" "interest_by_region"
## [4] "interest_by_dma"      "interest_by_city"    "related_topics"
## [7] "related_queries"
```

```
dim(x$interest_over_time)
```

```
## [1] 209   7
```

```
hist(x$interest_over_time[,2], 10)
```



```
head(x$related_topics)
```

```
##   subject related_topics             value keyword category
## 1     100            top         Influenza     flu        0
## 2      19            top Influenza vaccine     flu        0
## 3      16            top           Symptom     flu        0
## 4      12            top   Swine influenza     flu        0
## 5       9            top             Death     flu        0
## 6       8            top       Common cold     flu        0
```

EDA

```
us_flu<-x$interest_over_time[,1:2]
str(us_flu)
```

```
## 'data.frame':    209 obs. of  2 variables:
##  $ date: POSIXct, format: "2004-01-01" "2004-02-01" ...
##  $ hits: int  12 12 6 5 3 3 3 4 6 29 ...
```

```
us_flu[,1]<-as.factor(us_flu[,1])
attach(us_flu)
fluts<-ts(hits,c(2004,1),c(2021,4),12)
str(fluts)
```

```
##  Time-Series [1:208] from 2004 to 2021: 12 12 6 5 3 3 3 4 6 29 ...
```

```
fluts
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2004  12  12   6   5   3   3   3   4   6  29  13  11
## 2005  10  13  10   5   4   4   5   6  10  43  32  16
## 2006  17  19  20  13  13   8   5   5   7  14  14  10
## 2007  10  11   7   5   4   4   3   4   5  13  12   8
## 2008   8  13   9   5   4   4   3   4   5  12  12   9
## 2009   8   8   7 100  86  36  57  44  52  85  55  21
## 2010  12   9   7   6   5   4   4   6   9  12  10  13
## 2011  16  11   8   6   5   4   4   4   8  11   9   8
## 2012   8   9   8   6   6   4   5   5   9  12  11  15
## 2013  33  13   9   8   6   4   4   5   9  14  11  13
## 2014  22  11   8   7   6   5   5   6  10  16  12  17
## 2015  18  20  13   9   7   6   5   6   9  14  11  10
## 2016  10  12  13   9   8   6   5   6  11  15  11  12
## 2017  15  16  12   9   7   6   6   9  12  16  14  19
## 2018  51  41  14  10   8   7   6   7  12  20  16  16
## 2019  20  23  16  11   9   8   7   8  13  21  18  22
## 2020  31  40  85  41  21  13  18  16  24  29  20  18
## 2021  17  11  10  10
```

```
frequency(fluts)
```

```
## [1] 12
```

```
cycle(fluts)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2004   1   2   3   4   5   6   7   8   9  10  11  12
## 2005   1   2   3   4   5   6   7   8   9  10  11  12
## 2006   1   2   3   4   5   6   7   8   9  10  11  12
## 2007   1   2   3   4   5   6   7   8   9  10  11  12
## 2008   1   2   3   4   5   6   7   8   9  10  11  12
## 2009   1   2   3   4   5   6   7   8   9  10  11  12
## 2010   1   2   3   4   5   6   7   8   9  10  11  12
## 2011   1   2   3   4   5   6   7   8   9  10  11  12
## 2012   1   2   3   4   5   6   7   8   9  10  11  12
## 2013   1   2   3   4   5   6   7   8   9  10  11  12
## 2014   1   2   3   4   5   6   7   8   9  10  11  12
## 2015   1   2   3   4   5   6   7   8   9  10  11  12
## 2016   1   2   3   4   5   6   7   8   9  10  11  12
## 2017   1   2   3   4   5   6   7   8   9  10  11  12
## 2018   1   2   3   4   5   6   7   8   9  10  11  12
## 2019   1   2   3   4   5   6   7   8   9  10  11  12
## 2020   1   2   3   4   5   6   7   8   9  10  11  12
## 2021   1   2   3   4
```

```
summary(fluts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3.00    6.00   10.00   13.75   14.25  100.00
```

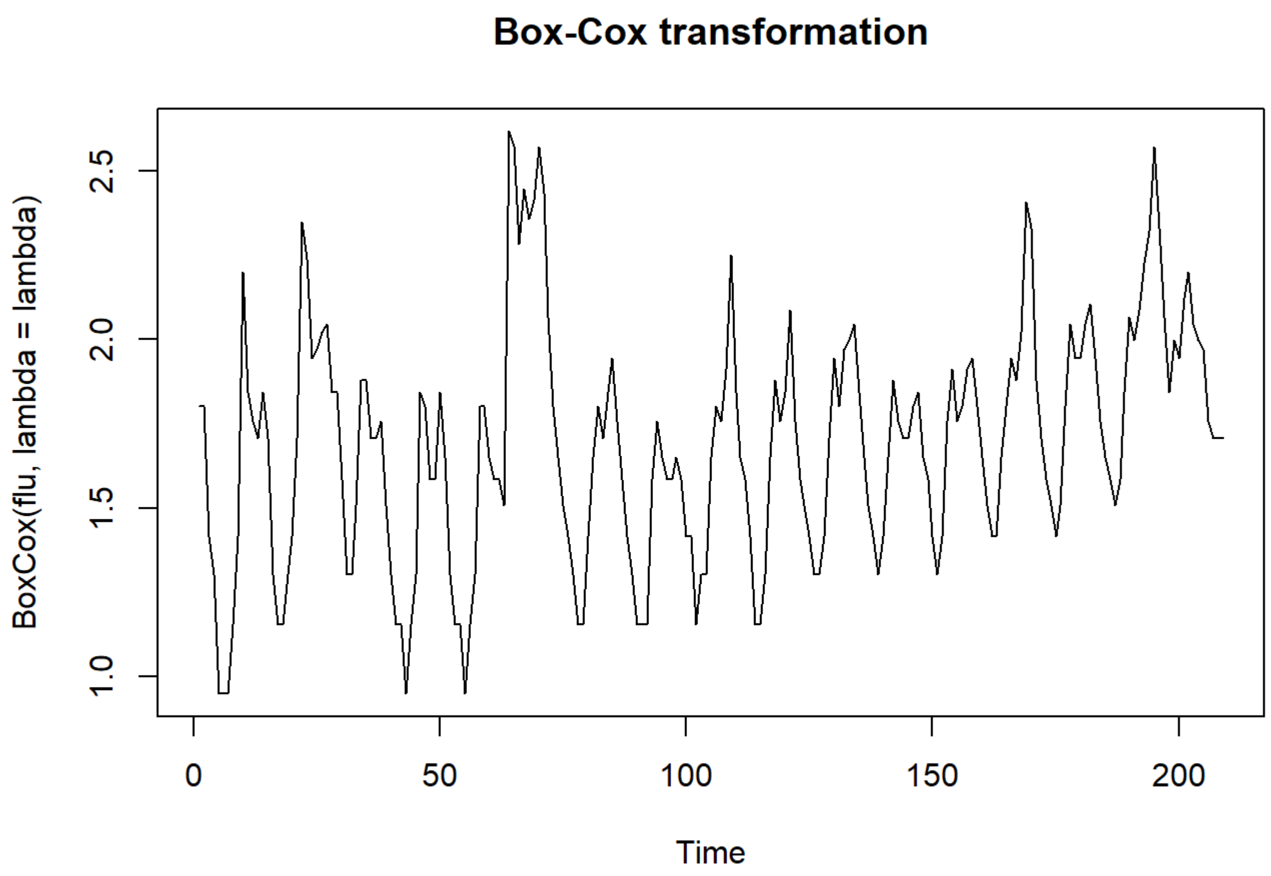
Box-cox

利用Box-Cox transformation，使轉換後的資料變異數齊一，更似常態分佈。 其中，計算出的lambda值為-0.2739011，並將轉換後的資料繪製成圖。

```
par(mfrow=c(1,1))
flu<-ts(x$interest_over_time[,2])
lambda <- BoxCox.lambda(flu)
print(lambda)
```

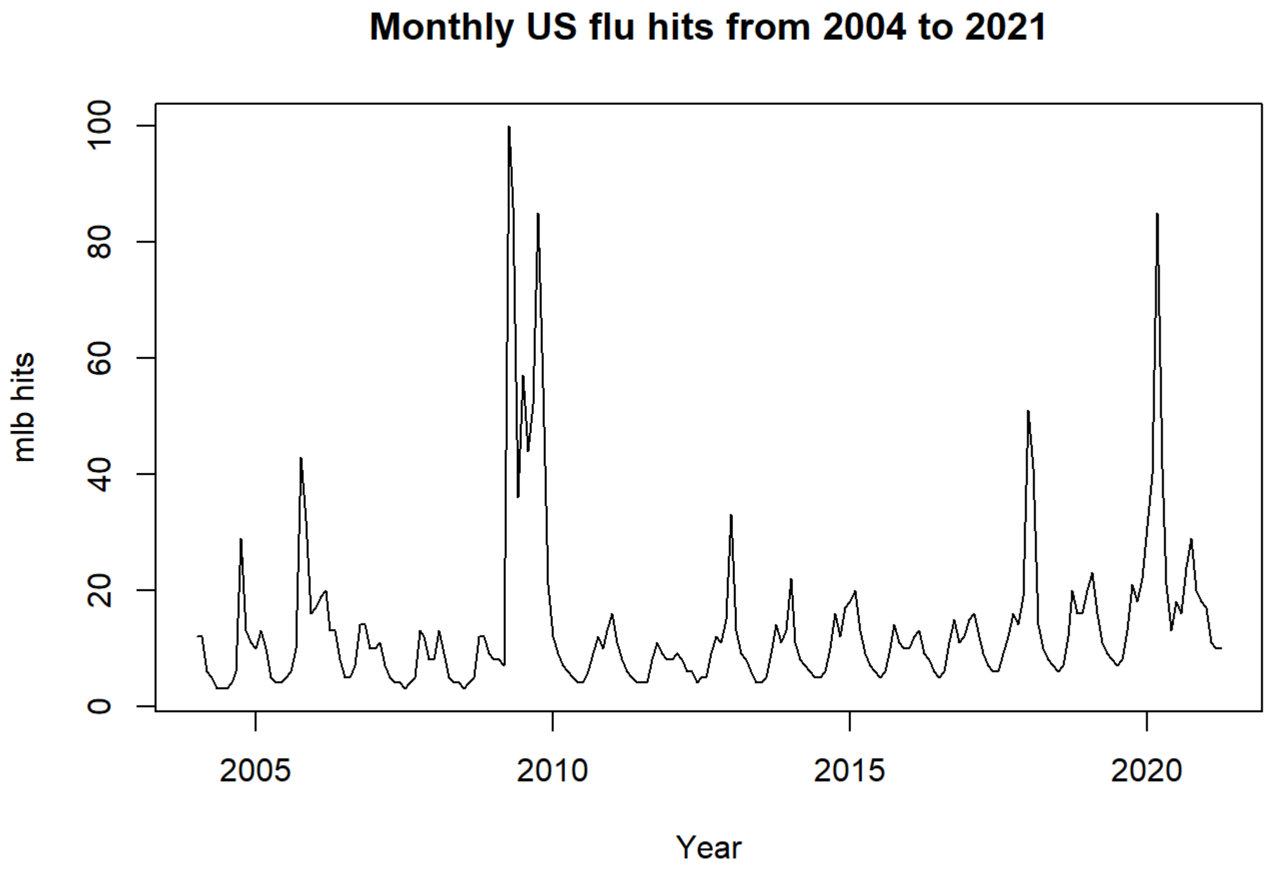
```
## [1] -0.2739011
```

```
plot.ts(BoxCox(flu, lambda = lambda), main='Box-Cox transformation')
```



TS-plot

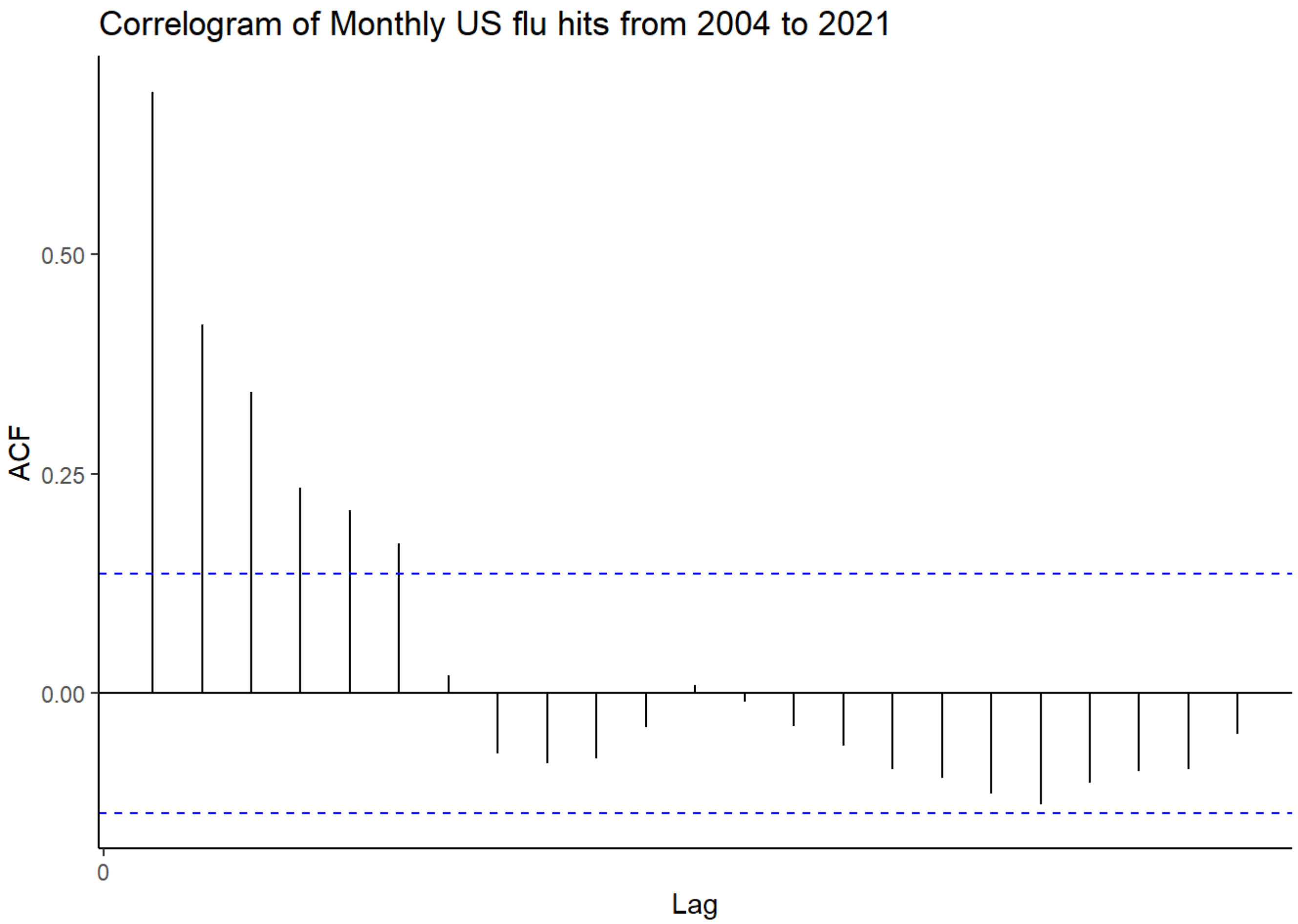
```
plot(fluts,xlab="Year", ylab = "mlb hits",
     main="Monthly US flu hits from 2004 to 2021")
```



ACF of fluts

繪製資料之ACF圖

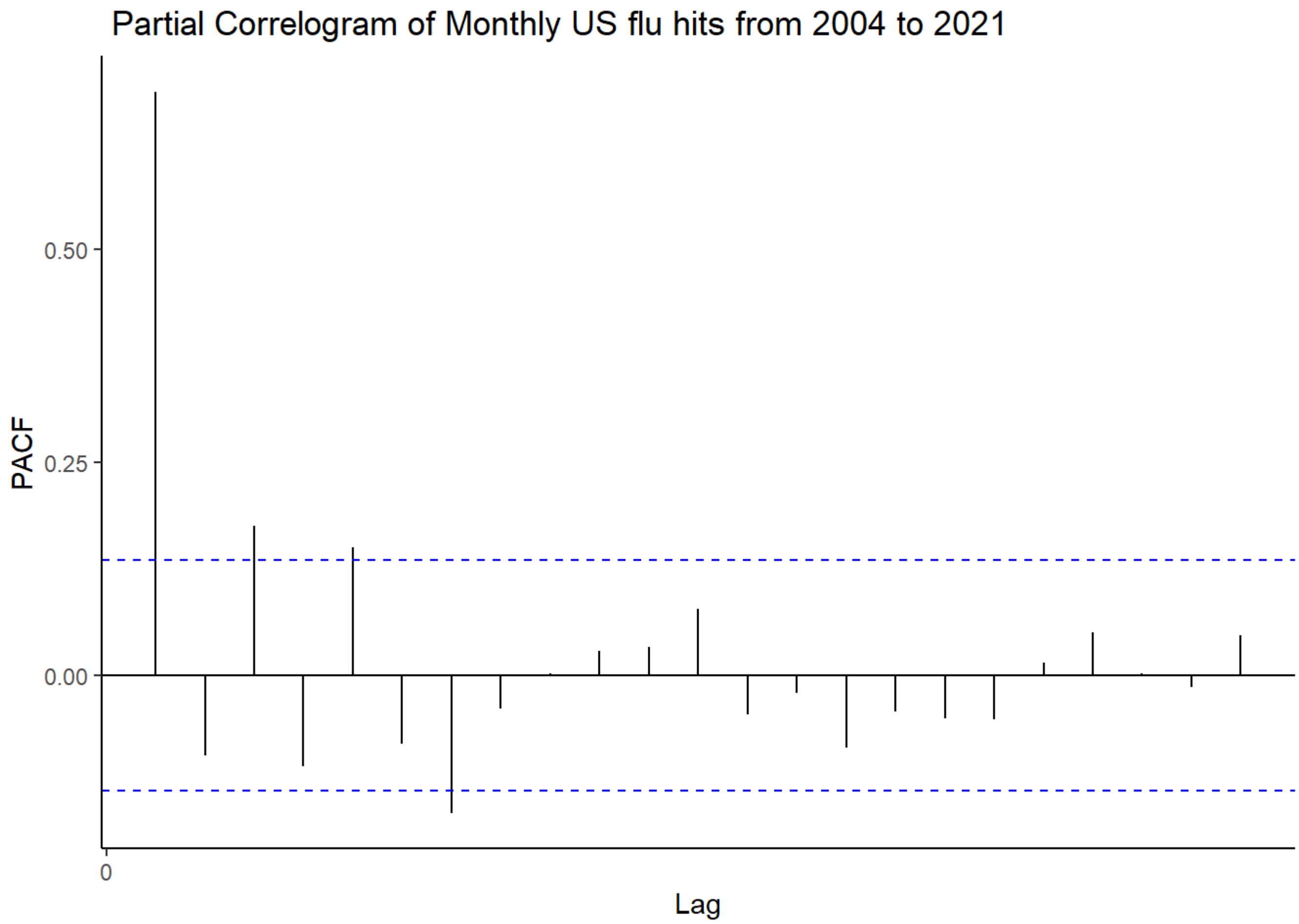
```
autoplot(acf(fluts,plot=FALSE))+
  labs(title="Correlogram of Monthly US flu hits from 2004 to 2021") + theme_classic()
```



PACF of fluts

繪製資料之PACF圖

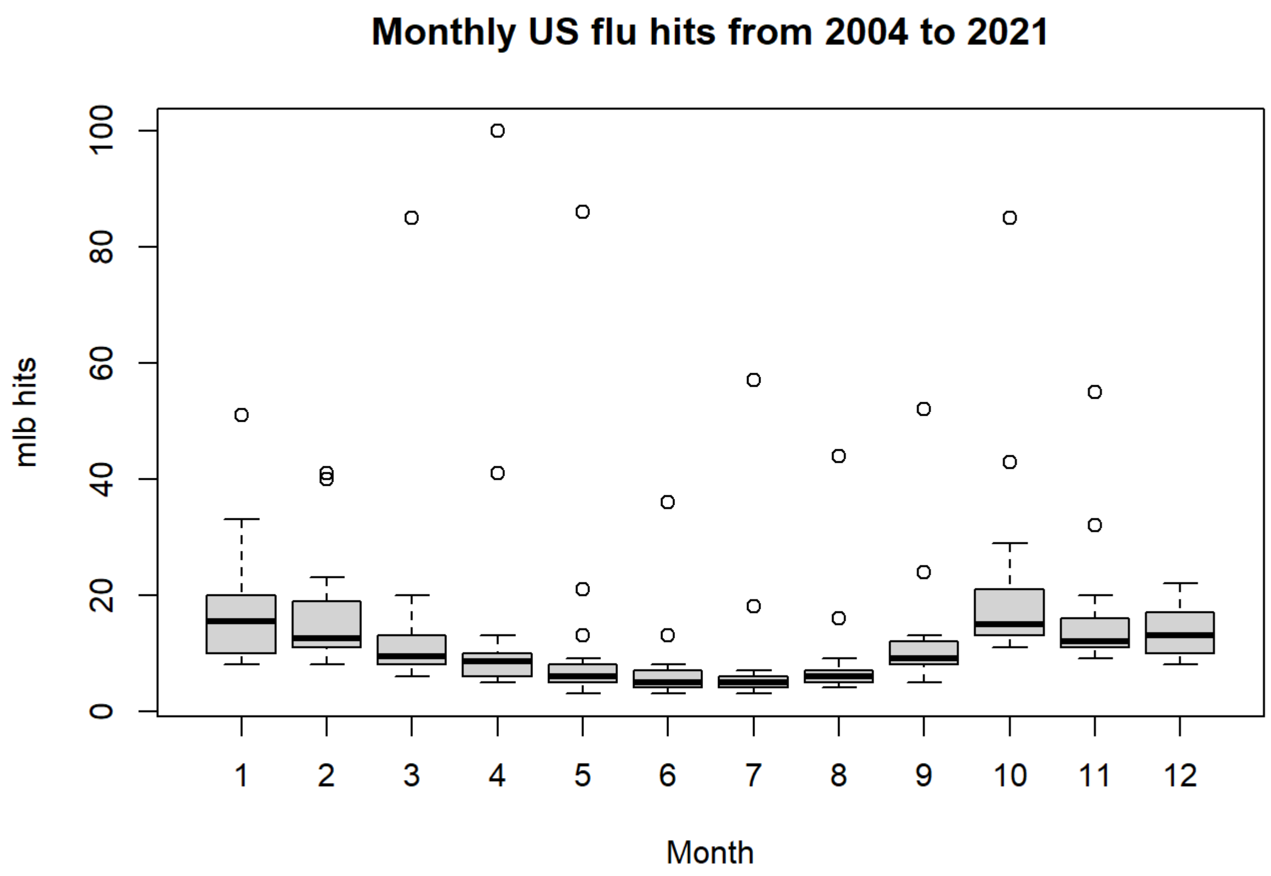
```
autoplot(pacf(fluts,plot=FALSE))+
  labs(title=" Partial Correlogram of Monthly US flu hits from 2004 to 2021") + theme_classic()
```



Boxplot

繪製資料之盒鬚圖，可看出2004~2021平均而言，關鍵字搜尋次數於10月份最多，7月最少。

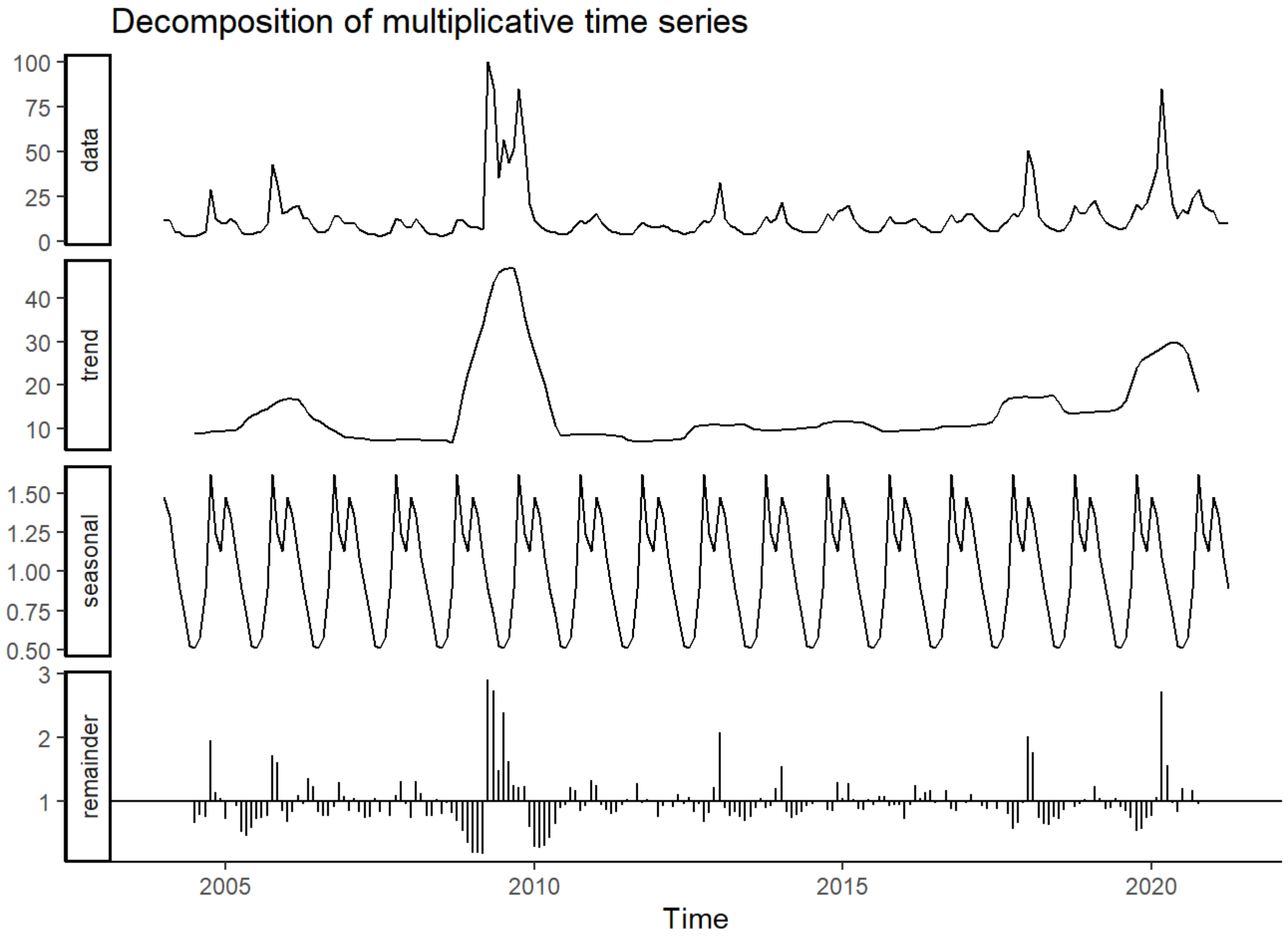
```
boxplot(fluts~cycle(fluts),xlab="Month", ylab = "mlb hits"
        ,main ="Monthly US flu hits from 2004 to 2021")
```



decomposition

將原始資料、趨勢、季節性、殘差分別繪製成圖。

```
decomp_fluts <- decompose(fluts,"multiplicative")
autoplot(decomp_fluts) + theme_classic()
```



Fitting Model method 1

Test stationality

以ADF test檢定平穩性，檢定結果顯著，此資料集為平穩序列。

```
adf.test(fluts)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  fluts
## Dickey-Fuller = -4.2005, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Fit arima

方法一將以auto.arima函數擬和模型。可得結果為ARIMA(1,0,2)。並可知AIC=1563.18。

```
arima_fluts <- auto.arima(fluts)
arima_fluts
```

```
## Series: fluts
## ARIMA(1,0,2) with non-zero mean
##
## Coefficients:
##          ar1     ma1      ma2     mean
##       0.7854  0.0329  -0.2904  13.6622
## s.e.  0.0841  0.1098   0.0943   2.3910
##
## sigma^2 estimated as 104.1:  log likelihood=-776.59
## AIC=1563.18   AICc=1563.48   BIC=1579.87
```

AIC

試在ARMA(i,0,0)，i從1到20中，尋找出AIC最小值。可得結果為AIC=1566.922。

```
out = matrix(0,20,4)
z = x$interest_over_time[,2]
for (i in 1:20){
  fit = arima(z, order=c(i,0,0), method="ML")
  out[i,] = c(i, fit$loglik, fit$aic, BIC(fit))
}
colnames(out) = c("p","loglik","AIC","BIC")
out = as.data.frame(out)
head(out)
```

```
##   p    loglik      AIC      BIC
## 1 1 -785.5620 1577.124 1587.151
## 2 2 -784.6550 1577.310 1590.679
## 3 3 -781.4034 1572.807 1589.518
## 4 4 -780.2149 1572.430 1592.484
## 5 5 -777.8449 1569.690 1593.086
## 6 6 -777.1783 1570.357 1597.095
```

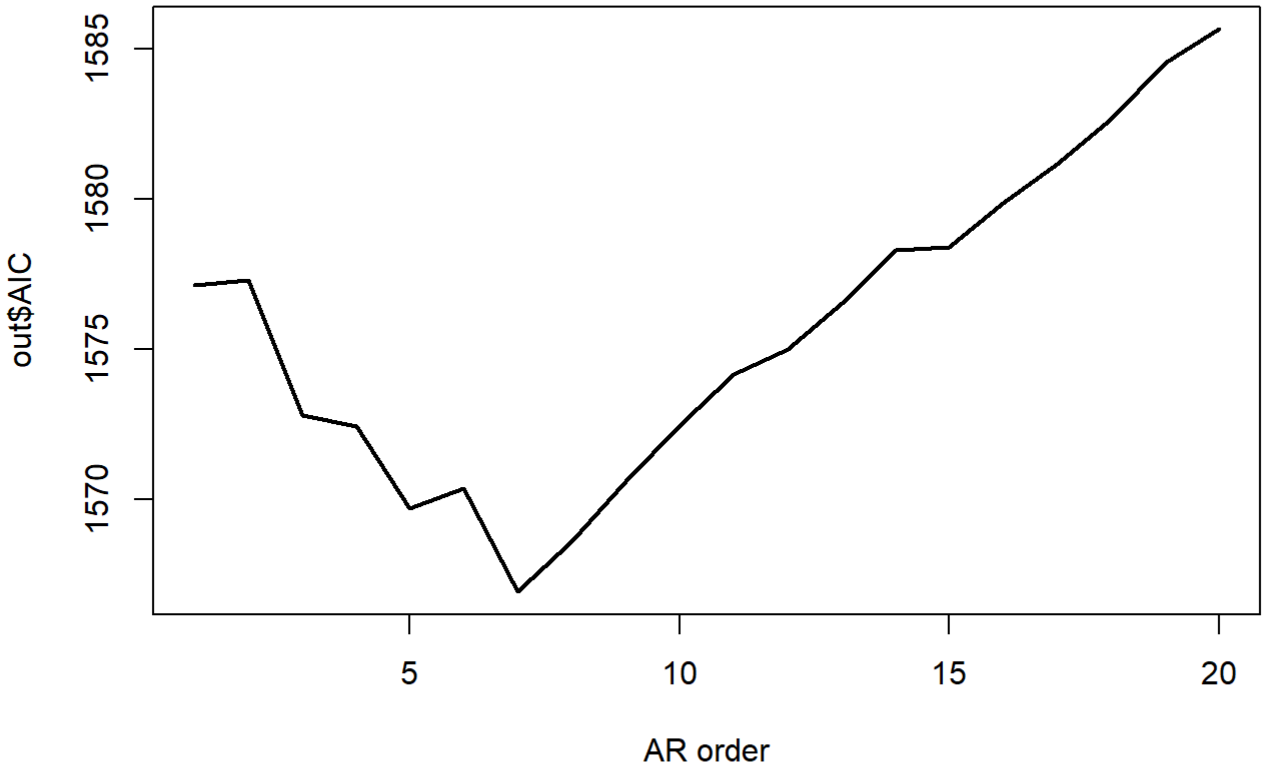
```
min(out$AIC)
```

```
## [1] 1566.922
```

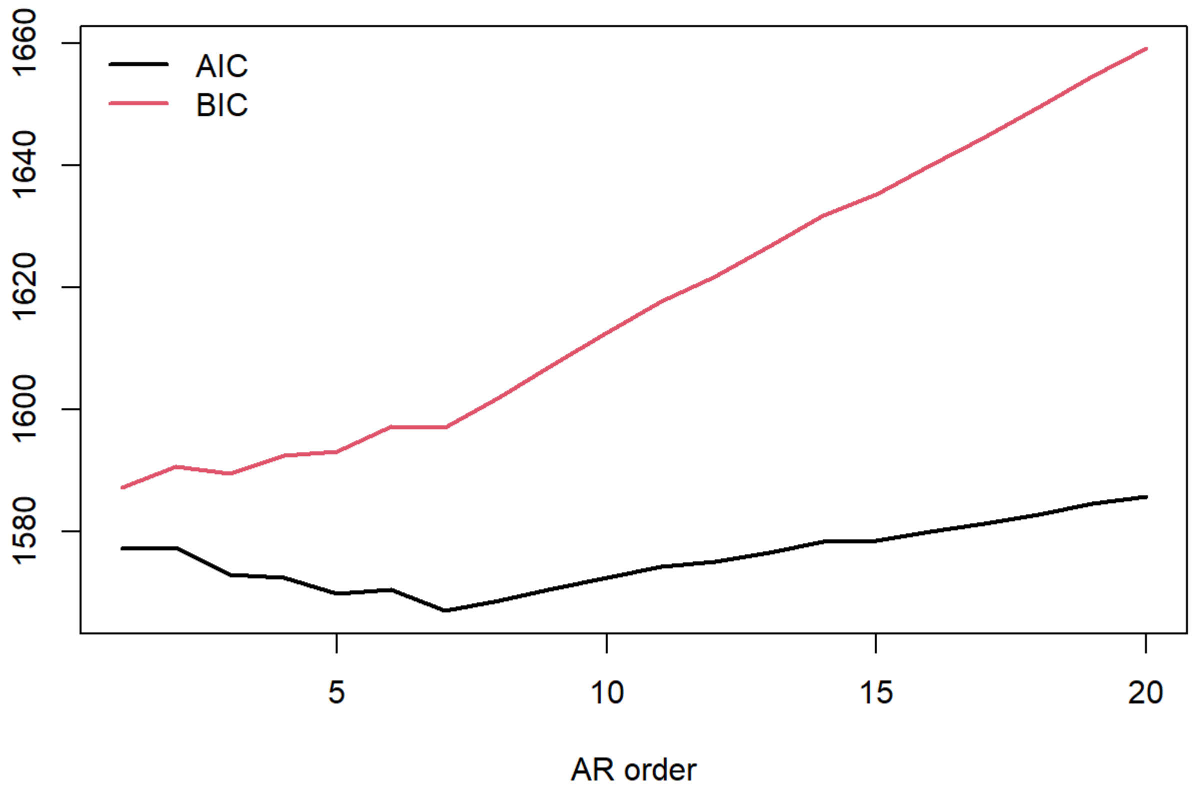
AIC plot

繪製AIC curve，可知在ARMA(7,0,0)時，AIC值達最小。可看出其值大於方才由auto.arima函數擬和出模型的AIC值。推測原因為真實模型不在candidate model中，故無法藉由此information criteria搜尋至最佳模型。

```
ts.plot(out$AIC, xlab="AR order", lwd=2) # plot log-likelihood v.s. AR order
```



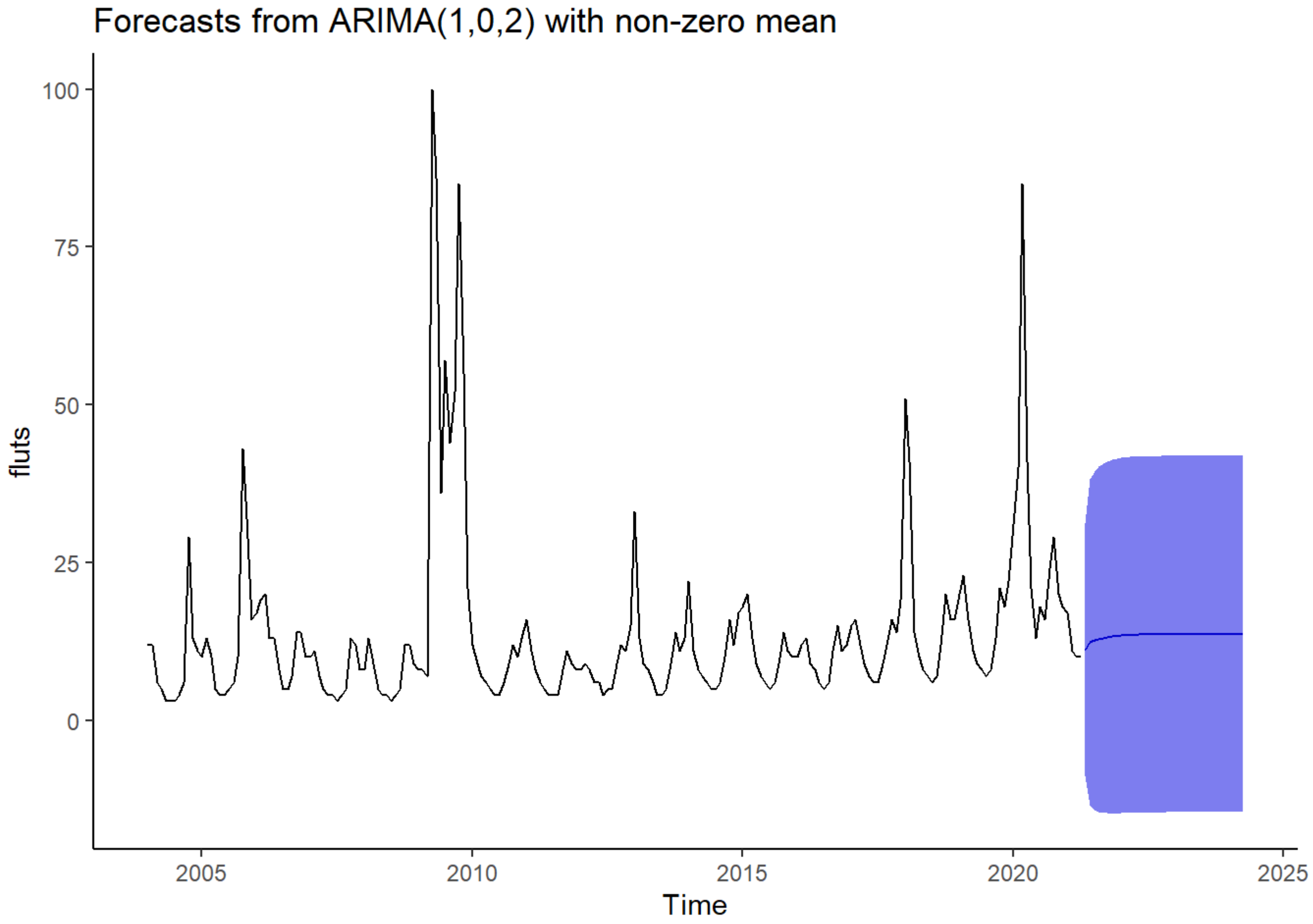
```
ts.plot(out[,3:4], col=1:2, lwd=2, xlab="AR order") #plot AIC and BIC v.s. AR order
legend("topleft", legend=c("AIC","BIC"), lty=1, col=1:2, lwd=2, bty="n")
```



Forcasting

繪製36步預測，並加上信賴區間。

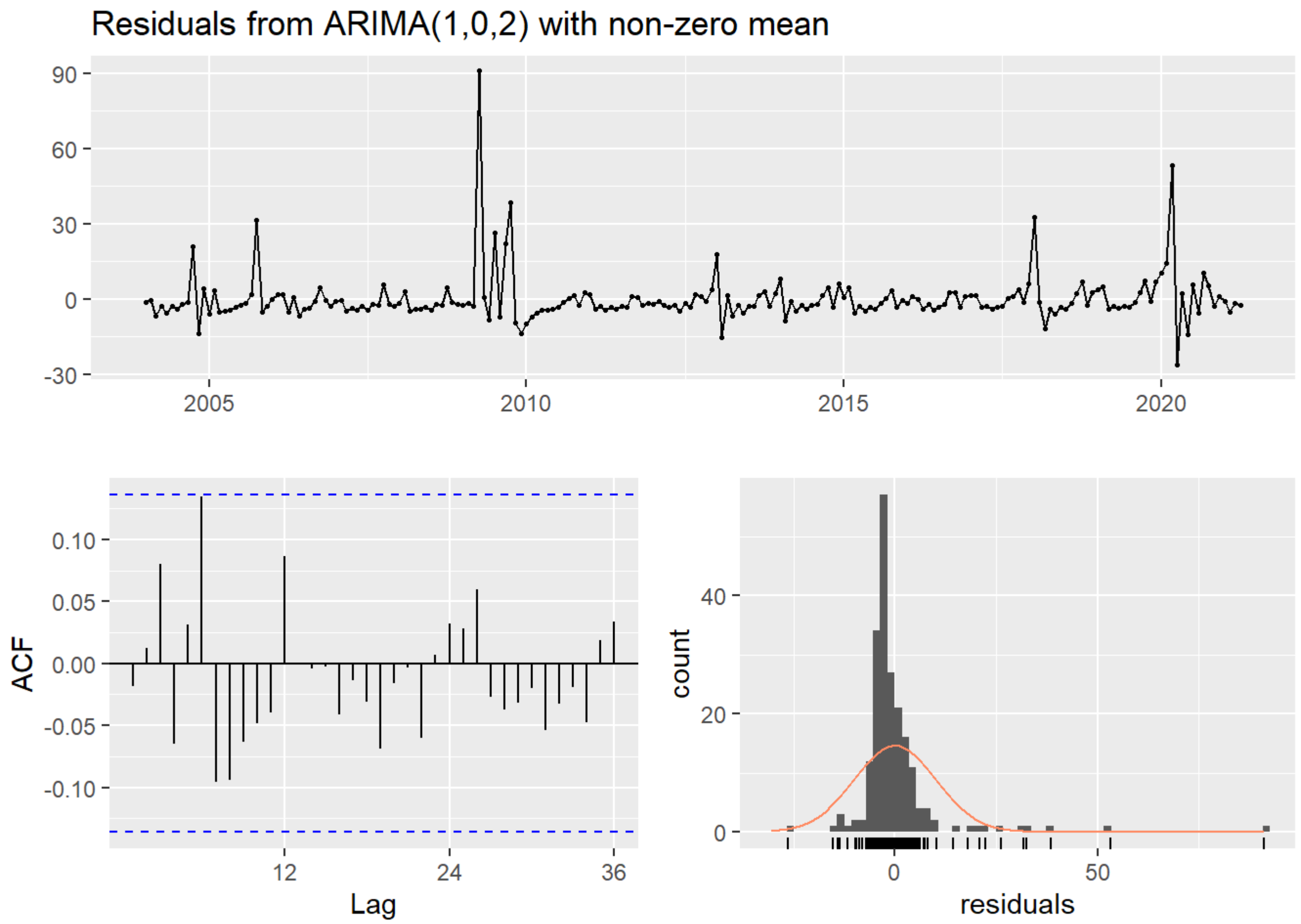
```
fore_fluts <- forecast(arima_fluts, level = c(95), h = 36)
autoplot(fore_fluts) + theme_classic()
```



Residual

對殘差作圖分析，可從ACF圖中看出直接落於95%信賴區間中，顯示此模型對相關結構作很好的描述。並可從直方圖可看出殘差呈現常態分配。

```
checkresiduals(arima_fluts)
```

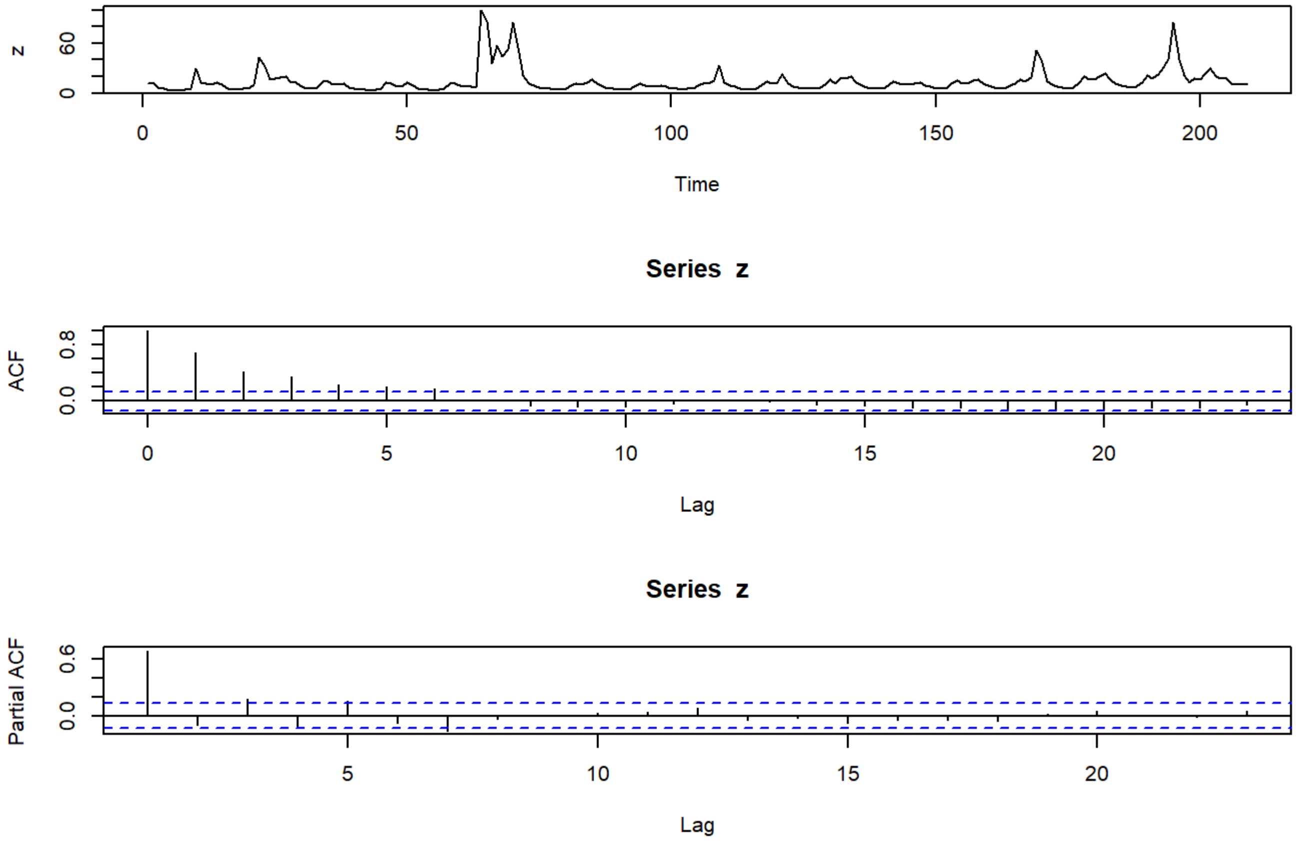


```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,2) with non-zero mean
## Q* = 16.711, df = 20, p-value = 0.6717
##
## Model df: 4.   Total lags used: 24
```

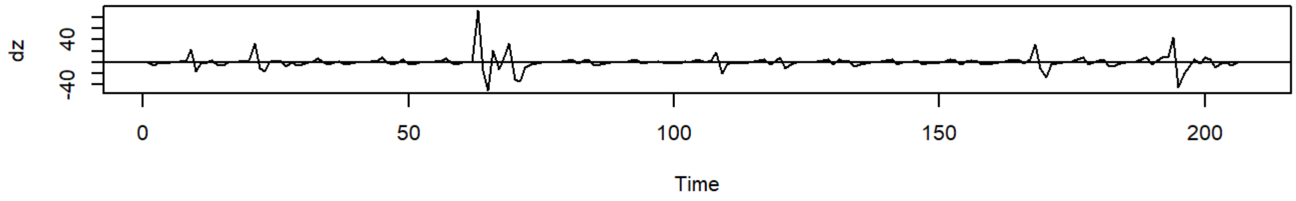
Fitting Model method 2 (Differencing)

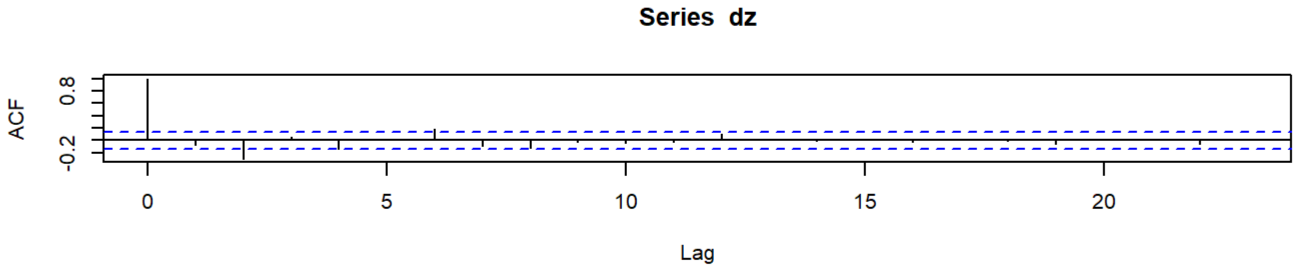
方法二將直接觀察ACF、PACF圖形，並找出應擬合之模型。首先先將資料作一次差分，並可藉繪製出的ACF、PACF圖形看出，ACF從lag2處開始tailoff，對應至AR(2)模型。且PACF從lag1開始cutoff，對應至MA(1)模型。另外，對資料作季節性差分，可看出其ACF從lag2處開始tailoff，對應至AR(2)模型。且PACF從lag1開始cutoff，對應至MA(1)模型。故最後選擇SARIMA(2,1,1)*(2,1,1)，其中週期為12。並可從殘差之ACF圖看出似white noise。

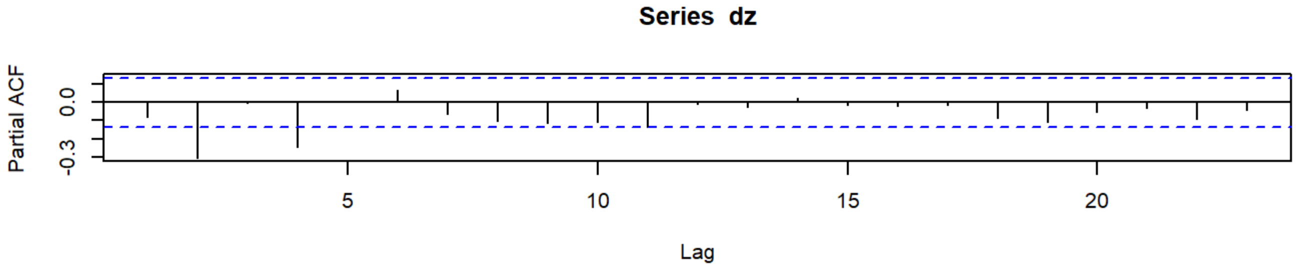
```
#original data
{par(mfrow=c(3,1))
  z = x$interest_over_time[,2]
  par(mfrow=c(3,1))
  ts.plot(z)
  acf(z)
  pacf(z)}
```



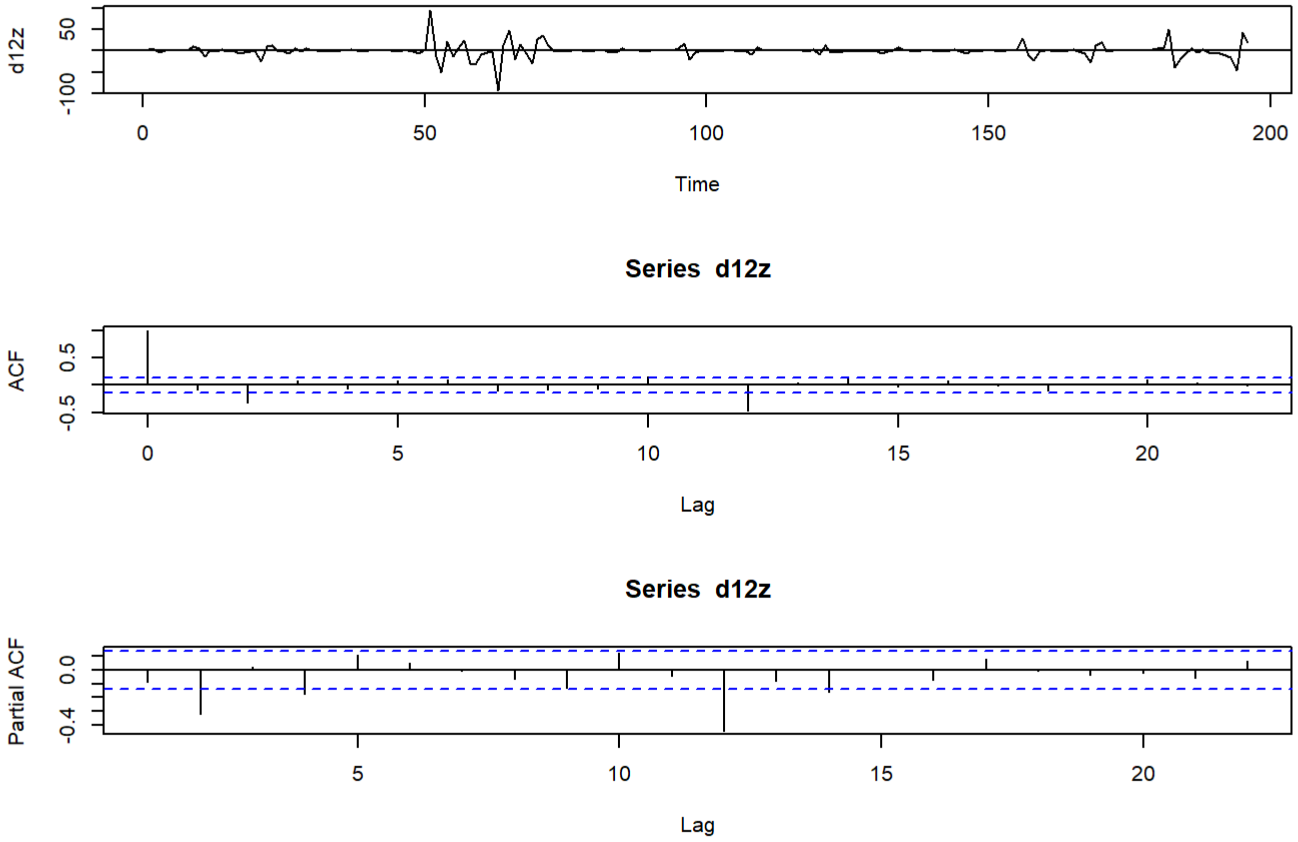
```
#diff
dz<-diff(z)
{par(mfrow=c(3,1))
  {ts.plot(dz)
    abline(h=mean(dz))
}
  acf(dz)
  pacf(dz)}
```





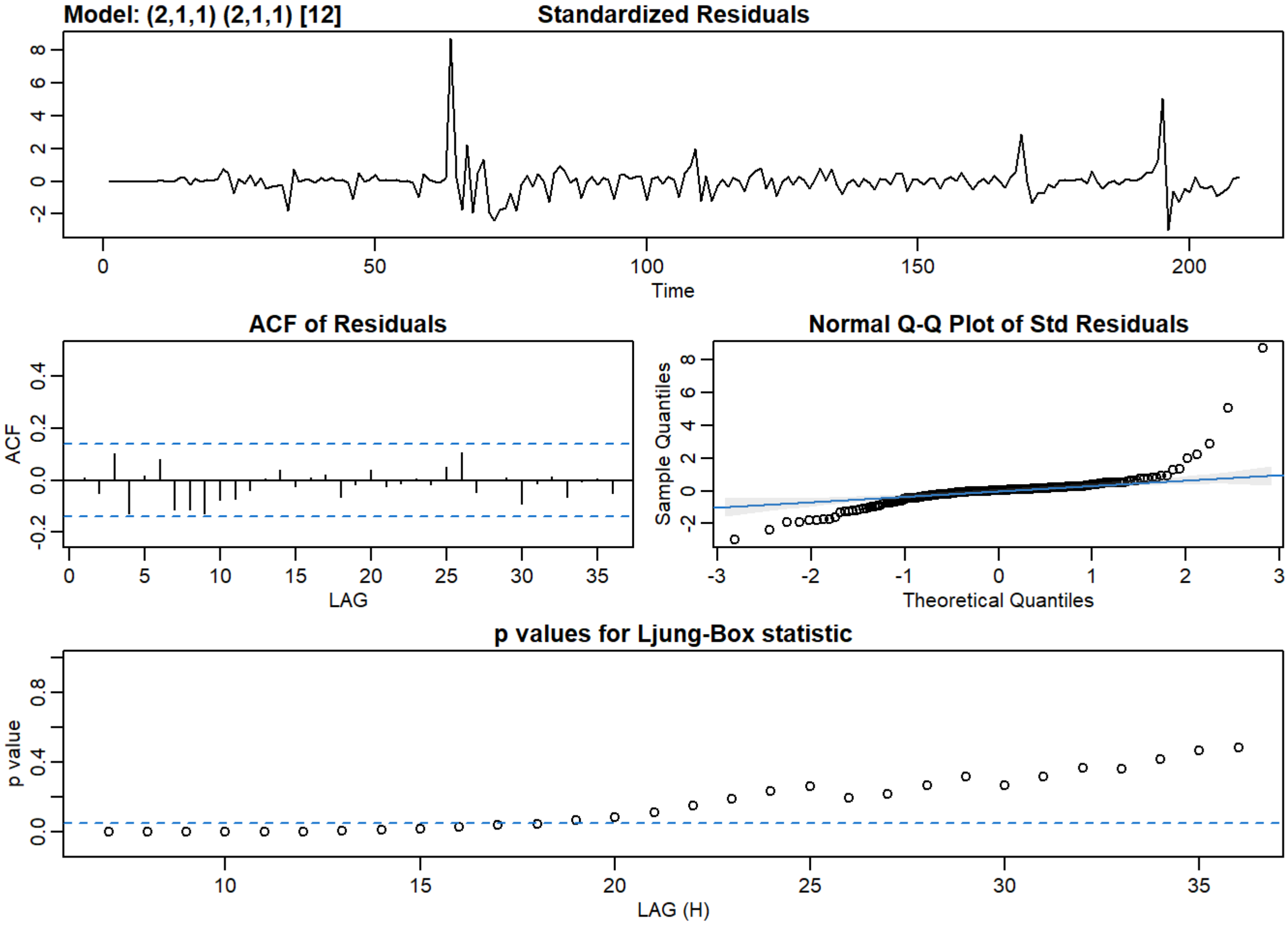


```
#seasonal diff
d12z<-diff(dz,12)
{par(mfrow=c(3,1))
  {ts.plot(d12z)
    abline(h=mean(d12z))
}
  acf(d12z)
  pacf(d12z)}
```



```
par(mfrow=c(1,1))
#choose model
sarima(z, 2,1,1,2,1,1,12)
```

```
## initial  value 2.787462 
## iter   2 value 2.510333
## iter   3 value 2.468839
## iter   4 value 2.453574
## iter   5 value 2.436167
## iter   6 value 2.430294
## iter   7 value 2.428824
## iter   8 value 2.424518
## iter   9 value 2.420875
## iter  10 value 2.419984
## iter  11 value 2.419352
## iter  12 value 2.419084
## iter  13 value 2.417637
## iter  14 value 2.415000
## iter  15 value 2.412662
## iter  16 value 2.410838
## iter  17 value 2.409285
## iter  18 value 2.406553
## iter  19 value 2.404987
## iter  20 value 2.401566
## iter  21 value 2.399896
## iter  22 value 2.399504
## iter  23 value 2.399209
## iter  24 value 2.399205
## iter  25 value 2.399205
## iter  25 value 2.399205
## iter  25 value 2.399205
## final  value 2.399205 
## converged
## initial  value 2.430192 
## iter   2 value 2.427824
## iter   3 value 2.424397
## iter   4 value 2.422959
## iter   5 value 2.421408
## iter   6 value 2.420757
## iter   7 value 2.420561
## iter   8 value 2.420435
## iter   9 value 2.420255
## iter  10 value 2.420015
## iter  11 value 2.419846
## iter  12 value 2.419734
## iter  13 value 2.418498
## iter  14 value 2.418181
## iter  15 value 2.417389
## iter  16 value 2.417228
## iter  17 value 2.416843
## iter  18 value 2.415889
## iter  19 value 2.415228
## iter  20 value 2.413953
## iter  21 value 2.413824
## iter  22 value 2.413806
## iter  23 value 2.413769
## iter  24 value 2.413764
## iter  25 value 2.413764
## iter  25 value 2.413764
## iter  25 value 2.413764
## final  value 2.413764
## converged
```

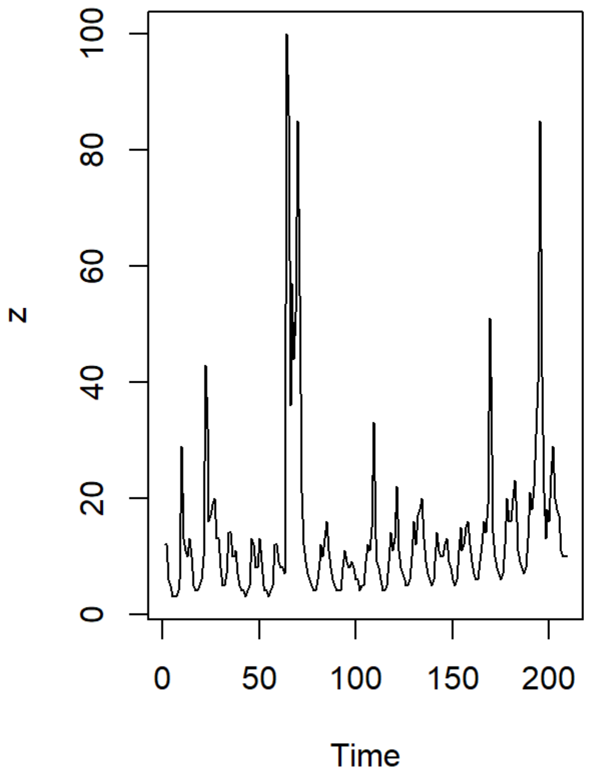


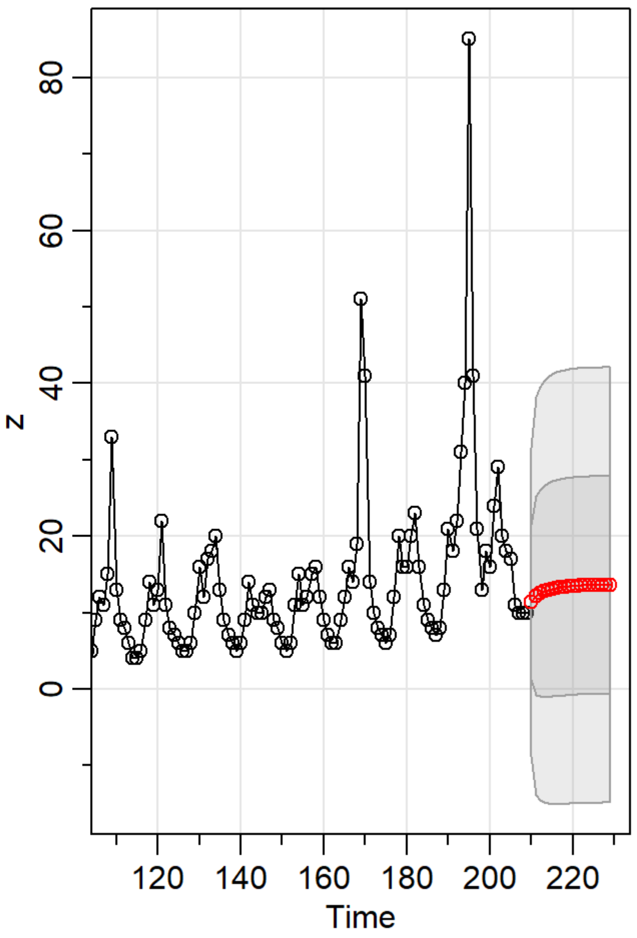
```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2      ma1     sar1     sar2     sma1
##       0.3144  -0.2502  -0.4723  -0.0311  -0.0435  -0.8556
## s.e.  0.2379   0.0888   0.2503   0.0992   0.0922   0.0885
##
## sigma^2 estimated as 114.3:  log likelihood = -751.21,  aic = 1516.42
##
## $degrees_of_freedom
## [1] 190
##
## $ttable
##      Estimate     SE t.value p.value
## ar1    0.3144 0.2379  1.3214  0.1879
## ar2   -0.2502 0.0888 -2.8186  0.0053
## ma1   -0.4723 0.2503 -1.8869  0.0607
## sar1  -0.0311 0.0992 -0.3139  0.7539
## sar2  -0.0435 0.0922 -0.4725  0.6371
## sma1  -0.8556 0.0885 -9.6682  0.0000
##
## $AIC
## [1] 7.325697
##
## $AICc
## [1] 7.327726
##
## $BIC
## [1] 7.436551
```

Forecast

同時繪製由方法一及方法二模型產生出的20步預測值。

```
par(mfrow=c(1,2))
{ts.plot(z)
  sarima.for(z, 20, 1,0,2)}
```





```
## $pred
## Time Series:
## Start = 210
## End = 229
## Frequency = 1
##  [1] 11.37076 12.20546 12.51402 12.75635 12.94667 13.09614 13.21353 13.30572
##  [9] 13.37812 13.43499 13.47964 13.51472 13.54226 13.56390 13.58089 13.59423
## [17] 13.60471 13.61294 13.61940 13.62448
##
## $se
## Time Series:
## Start = 210
## End = 229
## Frequency = 1
##  [1] 10.07817 13.02369 13.49976 13.78520 13.95835 14.06408 14.12890 14.16874
##  [9] 14.19325 14.20835 14.21765 14.22339 14.22693 14.22911 14.23045 14.23128
## [17] 14.23179 14.23211 14.23230 14.23242
```

Efficiency

將資料切分成訓練集及驗證集，並計算以兩擬合後模型，套用至驗證集上的rmse。可得兩方法結果相近。

```
# splitting data into train and valid sets
train = z[1:168]
valid = z[168:length(z)]

# training model
model = arima(train, order=c(1,0,2), method = 'ML')
model2 = arima(train, order=c(2,1,1), season = list(order=c(2,1,1), period=12), method = 'ML')

# model summary
summary(model)
```

```
##
## Call:
## arima(x = train, order = c(1, 0, 2), method = "ML")
##
## Coefficients:
##          ar1     ma1      ma2  intercept
##       0.8185  0.0418  -0.3578    12.3979
## s.e.  0.0765  0.1058   0.0913     2.7173
##
## sigma^2 estimated as 91:  log likelihood = -617.76,  aic = 1245.51
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.01789864 9.539399 4.484795 -24.16248 40.74724 0.9565272
##                     ACF1
## Training set -0.04178144
```

```
summary(model2)
```

```
##
## Call:
## arima(x = train, order = c(2, 1, 1), seasonal = list(order = c(2, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1      ar2     ma1    sar1    sar2     sma1
##       -0.6188  -0.3564  0.5716  0.0781  0.0196  -0.9999
## s.e.   0.1370   0.0770  0.1368  0.0847  0.0830   0.1615
##
## sigma^2 estimated as 92.69:  log likelihood = -585.84,  aic = 1185.69
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.02461056 9.247802 4.557193 -3.666891 37.91588 0.9719684
##                     ACF1
## Training set -0.01871893
```

```
# forecasting
forecast = predict(model,42)
forecast$pred
```

```
## Time Series:
## Start = 169
## End = 210
## Frequency = 1
##  [1] 18.43816 14.89398 14.44093 14.07012 13.76660 13.51818 13.31484 13.14841
##  [9] 13.01219 12.90069 12.80943 12.73473 12.67359 12.62355 12.58259 12.54906
## [17] 12.52162 12.49916 12.48078 12.46573 12.45341 12.44333 12.43508 12.42833
## [25] 12.42280 12.41827 12.41457 12.41154 12.40906 12.40703 12.40537 12.40401
## [33] 12.40289 12.40198 12.40124 12.40062 12.40012 12.39972 12.39938 12.39911
## [41] 12.39888 12.39870
```

```
forecast2 = predict(model2,42)
forecast2$pred
```

```
## Time Series:
## Start = 169
## End = 210
## Frequency = 1
##  [1] 18.46003 15.73244 14.62412 17.23150 14.91558 10.98876 11.84733 12.08542
##  [9] 15.56825 25.28455 19.91839 17.43601 18.97711 17.30804 14.66086 18.26928
## [17] 16.16682 11.67784 12.74976 12.85881 16.26557 26.45774 20.87633 17.87219
## [25] 19.51302 17.85488 15.14098 18.93932 16.84757 12.25647 13.36227 13.40700
## [33] 16.81716 27.15892 21.49455 18.30258 19.99210 18.35562 15.60627 19.43912
## [41] 17.35240 12.74227
```

```
# evaluation
rmse(valid, forecast$pred)
```

```
## [1] 15.85778
```

```
rmse(valid, forecast2$pred)
```

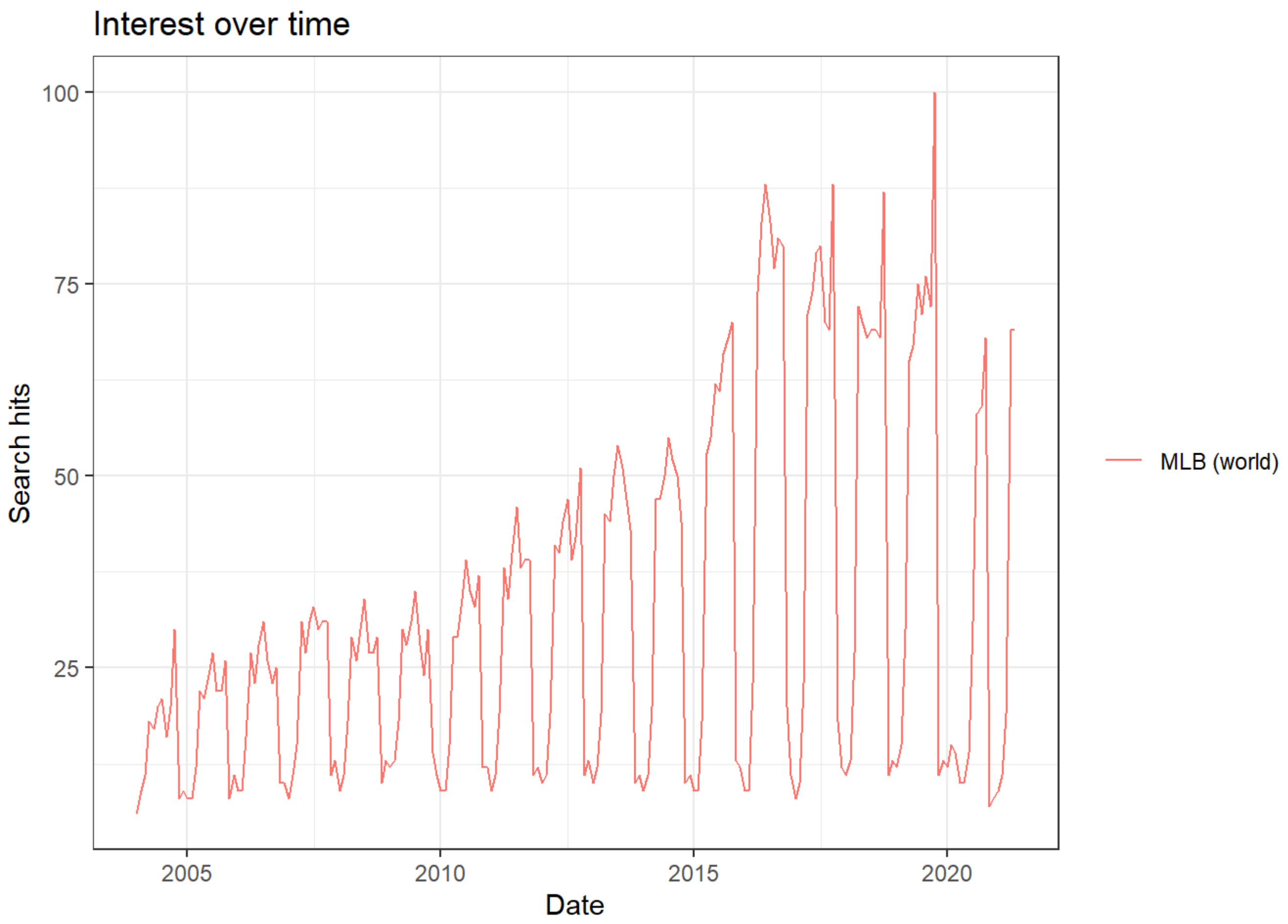
```
## [1] 14.48839
```

Dataset MLB

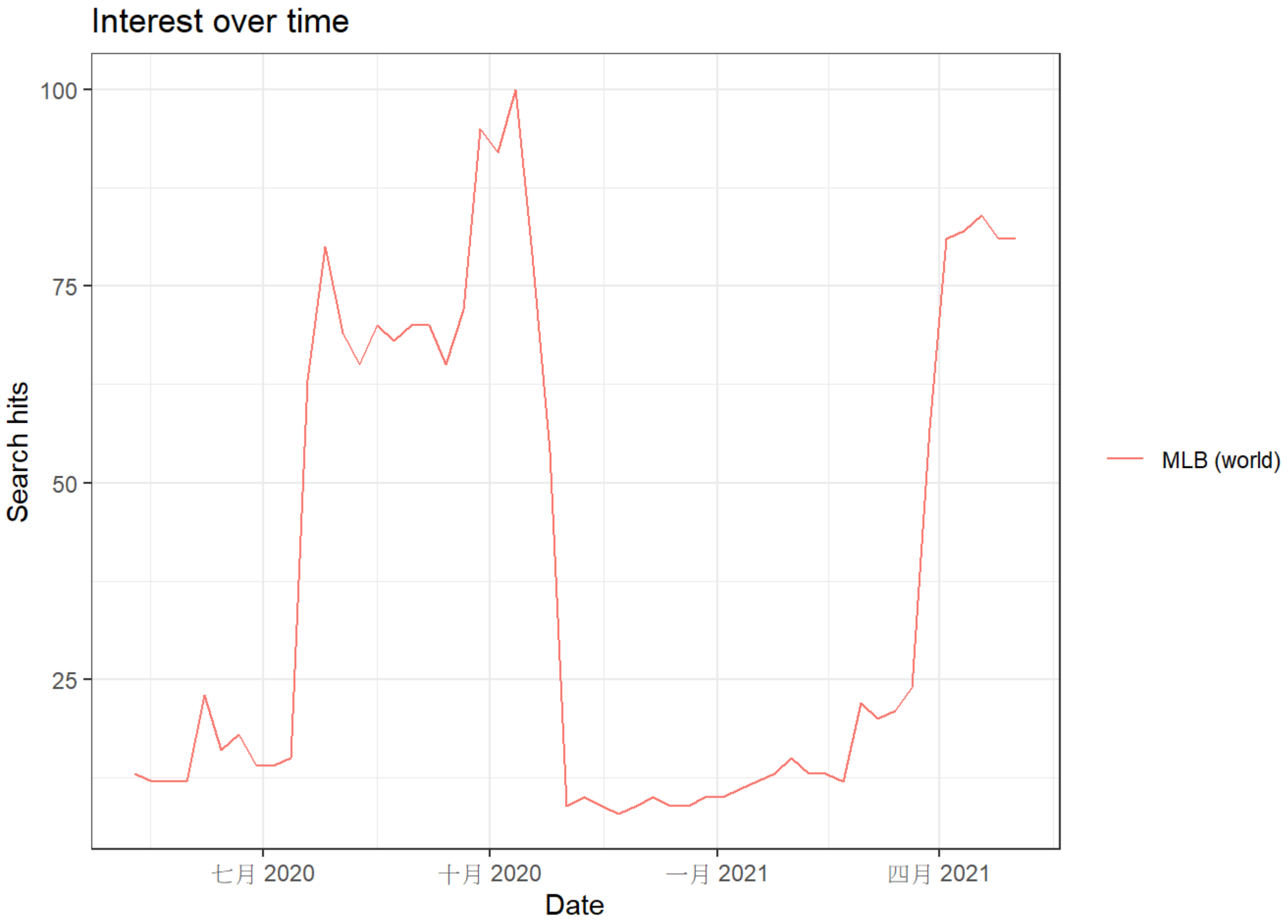
Basic information

此為關鍵字為MLB之資料集，並針對其作線圖及直方圖。

```
a = gtrends("MLB", time="all")
plot(a)
```



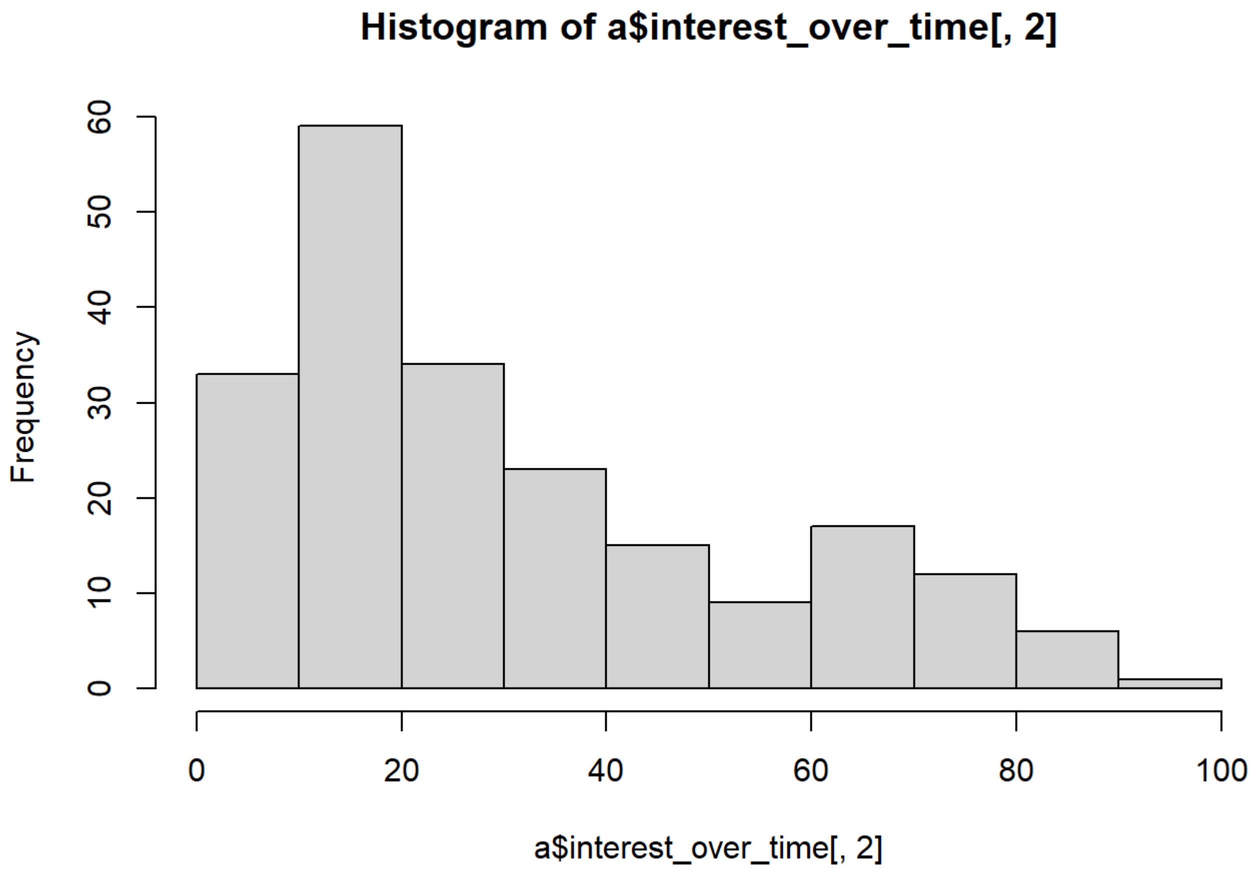
```
b = gtrends("MLB", time="today 12-m")
plot(b)
```



```
names(a)
```

```
## [1] "interest_over_time"   "interest_by_country" "interest_by_region"
## [4] "interest_by_dma"      "interest_by_city"    "related_topics"
## [7] "related_queries"
```

```
hist(a$interest_over_time[,2], 10)
```



```
head(a$related_topics)
```

```
##   subject related_topics                       value keyword category
## 1     100            top                         MLB     MLB        0
## 2       8            top                   Standings     MLB        0
## 3       4            top                        ESPN     MLB        0
## 4       4            top                        ESPN     MLB        0
## 5       4            top                         NBA     MLB        0
## 6       3            top ESPN Major League Baseball     MLB        0
```

EDA

```
us_mlb<-a$interest_over_time[,1:2]
str(us_mlb)
```

```
## 'data.frame':    209 obs. of  2 variables:
##  $ date: POSIXct, format: "2004-01-01" "2004-02-01" ...
##  $ hits: int  6 9 11 18 17 20 21 16 20 30 ...
```

```
us_mlb[,1]<-as.factor(us_mlb[,1])
attach(us_mlb)
```

```
## The following objects are masked from us_flu:
##
##     date, hits
```

```
mlbts<-ts(hits,c(2004,1),c(2021,4),12)
str(mlbts)
```

```
##  Time-Series [1:208] from 2004 to 2021: 6 9 11 18 17 20 21 16 20 30 ...
```

```
mlbts
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2004   6   9  11  18  17  20  21  16  20  30   8   9
## 2005   8   8  12  22  21  24  27  22  22  26   8  11
## 2006   9   9  16  27  23  28  31  26  23  25  10  10
## 2007   8  11  15  31  27  31  33  30  31  31  11  13
## 2008   9  11  18  29  26  30  34  27  27  29  10  13
## 2009  12  13  18  30  28  31  35  28  24  30  14  11
## 2010   9   9  15  29  29  34  39  35  33  37  12  12
## 2011   9  11  19  38  34  40  46  38  39  39  11  12
## 2012  10  11  19  41  40  44  47  39  42  51  11  13
## 2013  10  12  19  45  44  50  54  51  47  43  10  11
## 2014   9  11  20  47  47  50  55  52  50  44  10  11
## 2015   9   9  18  53  55  62  61  66  68  70  13  12
## 2016   9   9  23  74  83  88  83  77  81  80  20  11
## 2017   8  10  23  71  74  79  80  70  69  88  18  12
## 2018  11  13  27  72  70  68  69  69  68  87  11  13
## 2019  12  15  30  65  67  75  71  76  72 100  11  13
## 2020  12  15  14  10  10  14  31  58  59  68   7   8
## 2021   9  11  19  69
```

```
frequency(mlbts)
```

```
## [1] 12
```

```
cycle(mlbts)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2004   1   2   3   4   5   6   7   8   9  10  11  12
## 2005   1   2   3   4   5   6   7   8   9  10  11  12
## 2006   1   2   3   4   5   6   7   8   9  10  11  12
## 2007   1   2   3   4   5   6   7   8   9  10  11  12
## 2008   1   2   3   4   5   6   7   8   9  10  11  12
## 2009   1   2   3   4   5   6   7   8   9  10  11  12
## 2010   1   2   3   4   5   6   7   8   9  10  11  12
## 2011   1   2   3   4   5   6   7   8   9  10  11  12
## 2012   1   2   3   4   5   6   7   8   9  10  11  12
## 2013   1   2   3   4   5   6   7   8   9  10  11  12
## 2014   1   2   3   4   5   6   7   8   9  10  11  12
## 2015   1   2   3   4   5   6   7   8   9  10  11  12
## 2016   1   2   3   4   5   6   7   8   9  10  11  12
## 2017   1   2   3   4   5   6   7   8   9  10  11  12
## 2018   1   2   3   4   5   6   7   8   9  10  11  12
## 2019   1   2   3   4   5   6   7   8   9  10  11  12
## 2020   1   2   3   4   5   6   7   8   9  10  11  12
## 2021   1   2   3   4
```

```
summary(mlbts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    6.00   12.00   25.50   31.62   45.25  100.00
```

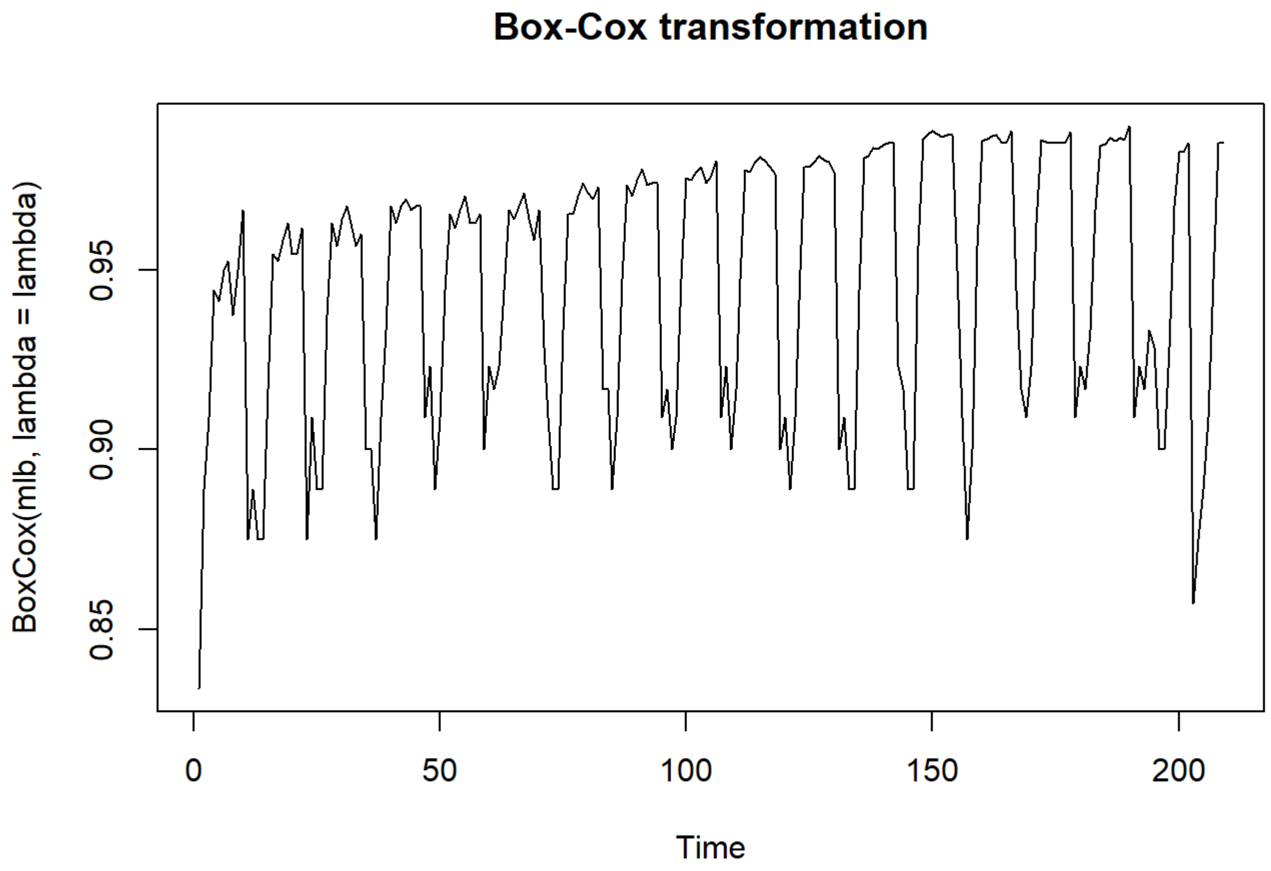
Box-cox

利用Box-Cox transformation，使轉換後的資料變異數齊一，更似常態分佈。 其中，計算出的lambda值為-0.9999242，並將轉換後的資料繪製成圖。

```
par(mfrow=c(1,1))
mlb<-ts(a$interest_over_time[,2])
lambda <- BoxCox.lambda(log(mlb))
print(lambda)
```

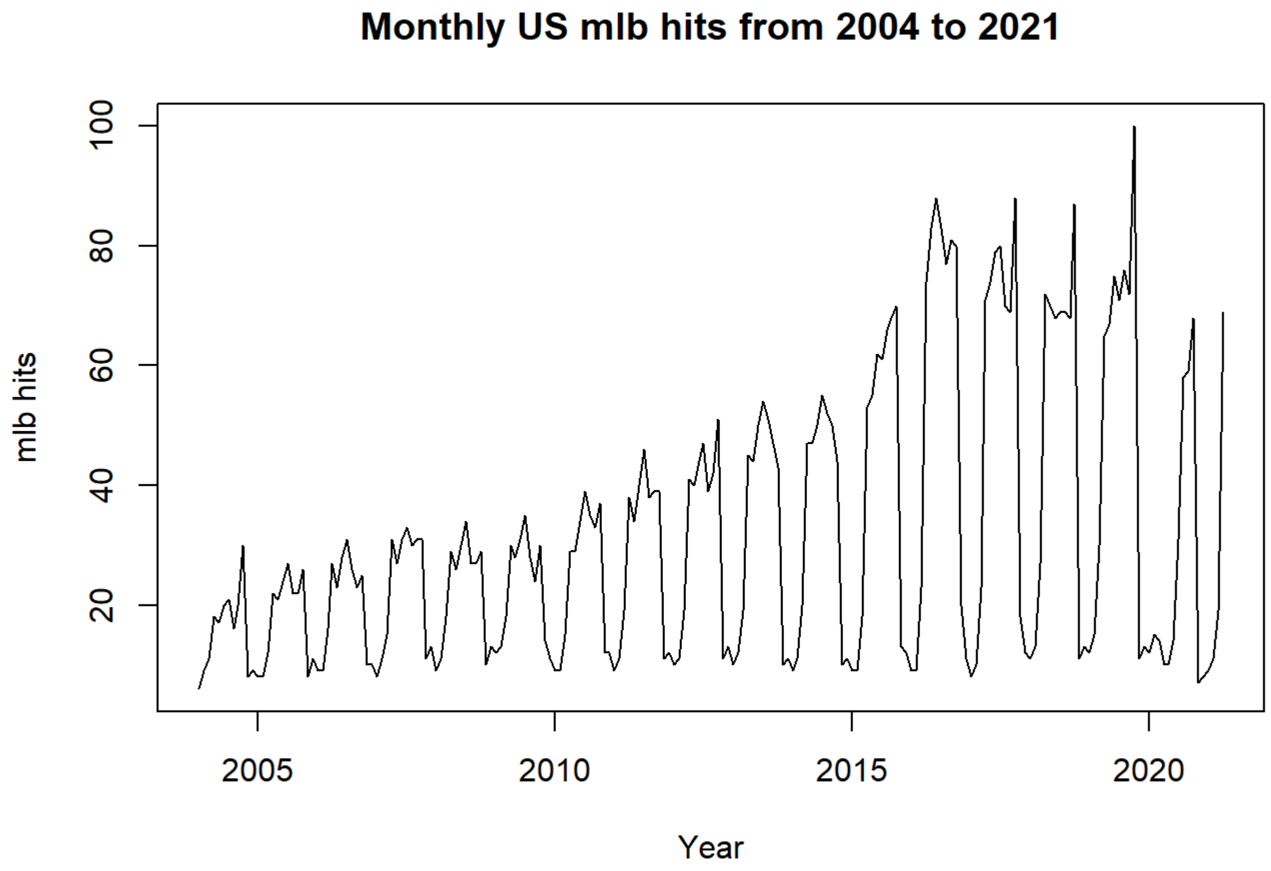
```
## [1] -0.9999242
```

```
plot.ts(BoxCox(mlb, lambda = lambda), main='Box-Cox transformation')
```



TS-plot

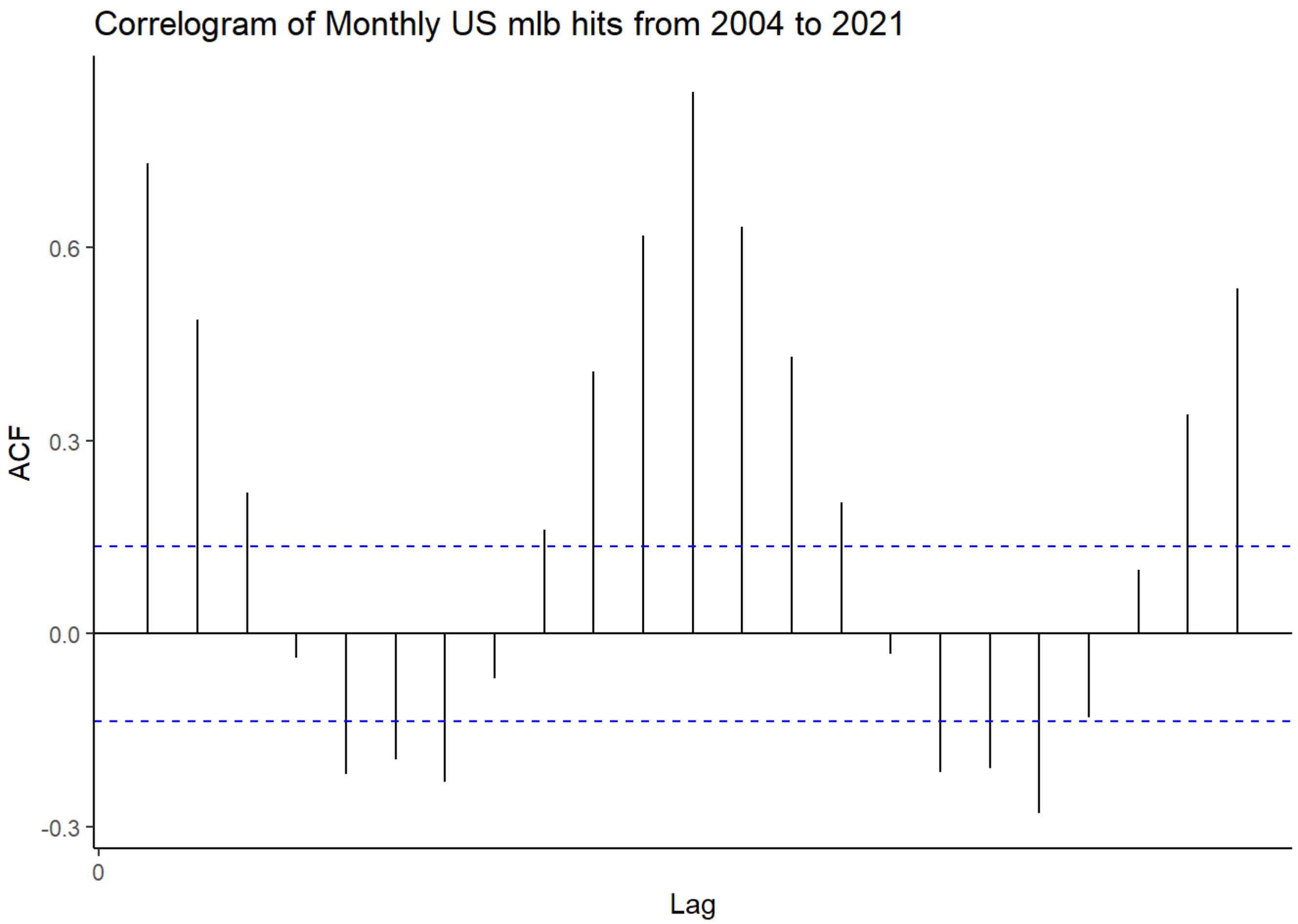
```
plot(mlbts,xlab="Year", ylab = "mlb hits",
    main="Monthly US mlb hits from 2004 to 2021")
```



ACF of fluts

繪製資料之ACF圖

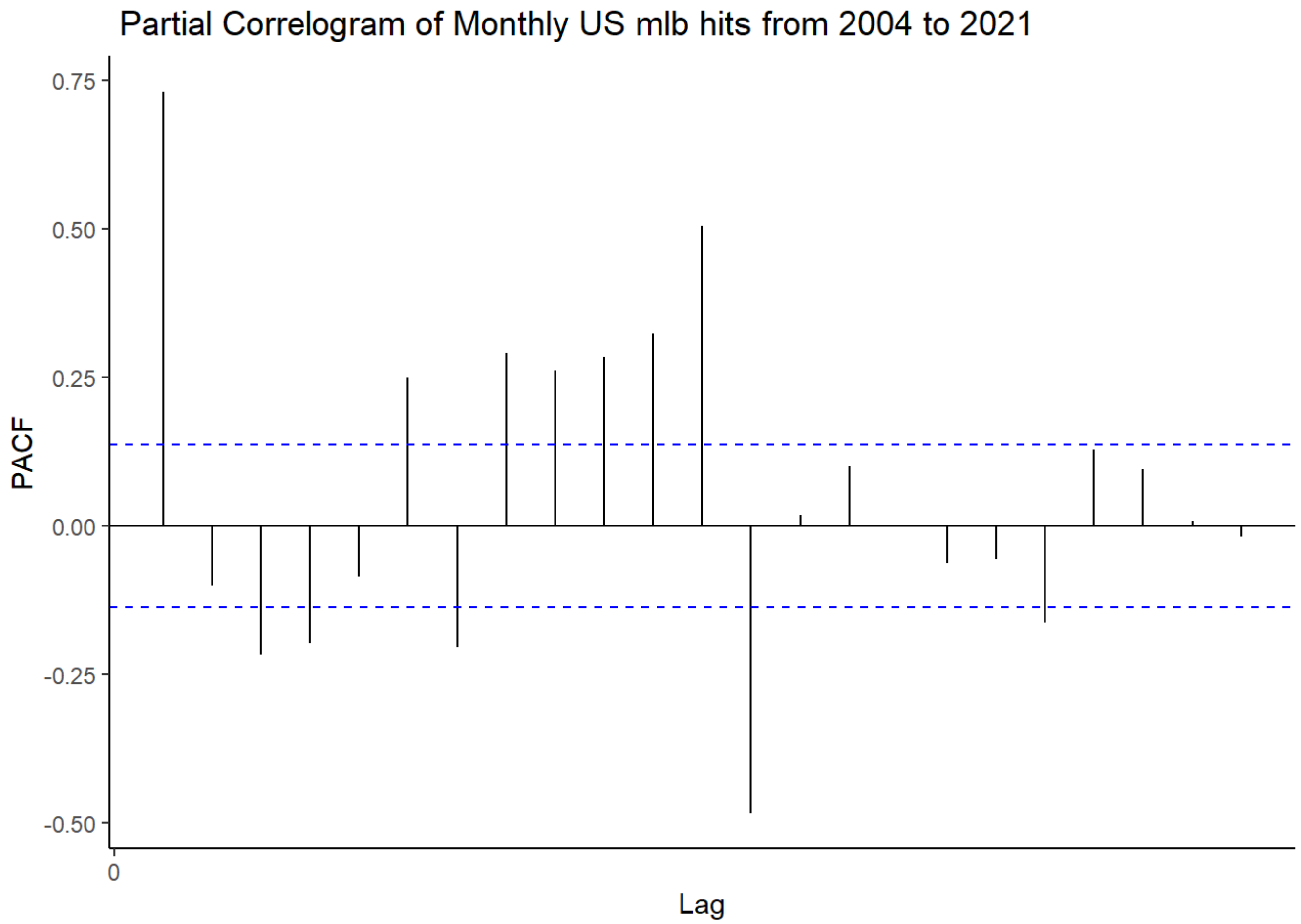
```
autoplot(acf(mlbts,plot=FALSE))+
  labs(title="Correlogram of Monthly US mlb hits from 2004 to 2021") + theme_classic()
```



PACF of fluts

繪製資料之PACF圖

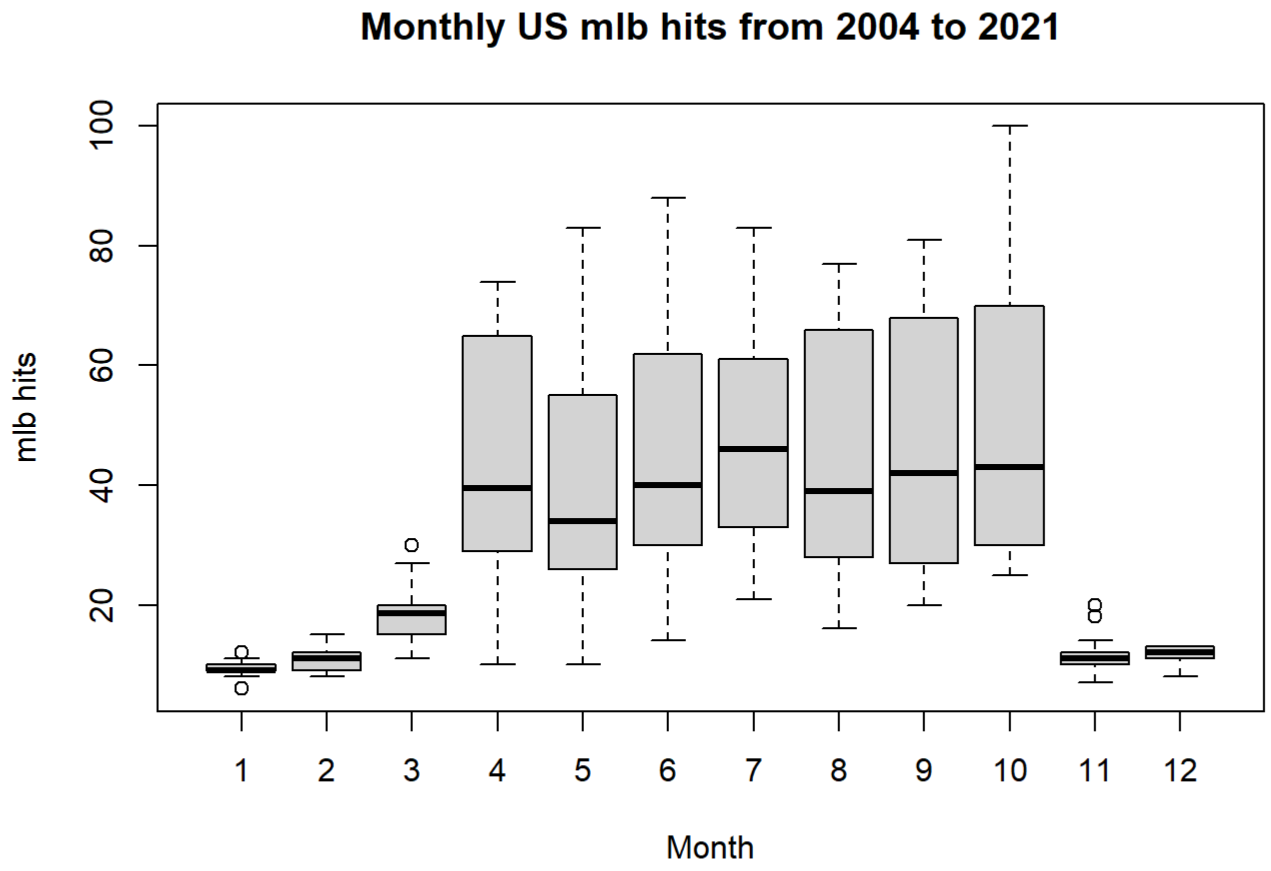
```
autoplot(pacf(mlbts,plot=FALSE))+
  labs(title=" Partial Correlogram of Monthly US mlb hits from 2004 to 2021") + theme_classic()
```



Boxplot

繪製資料之盒鬚圖，可看出2004$_{\text{2021平均而言，關鍵字搜尋次數集中於4}}$10月。

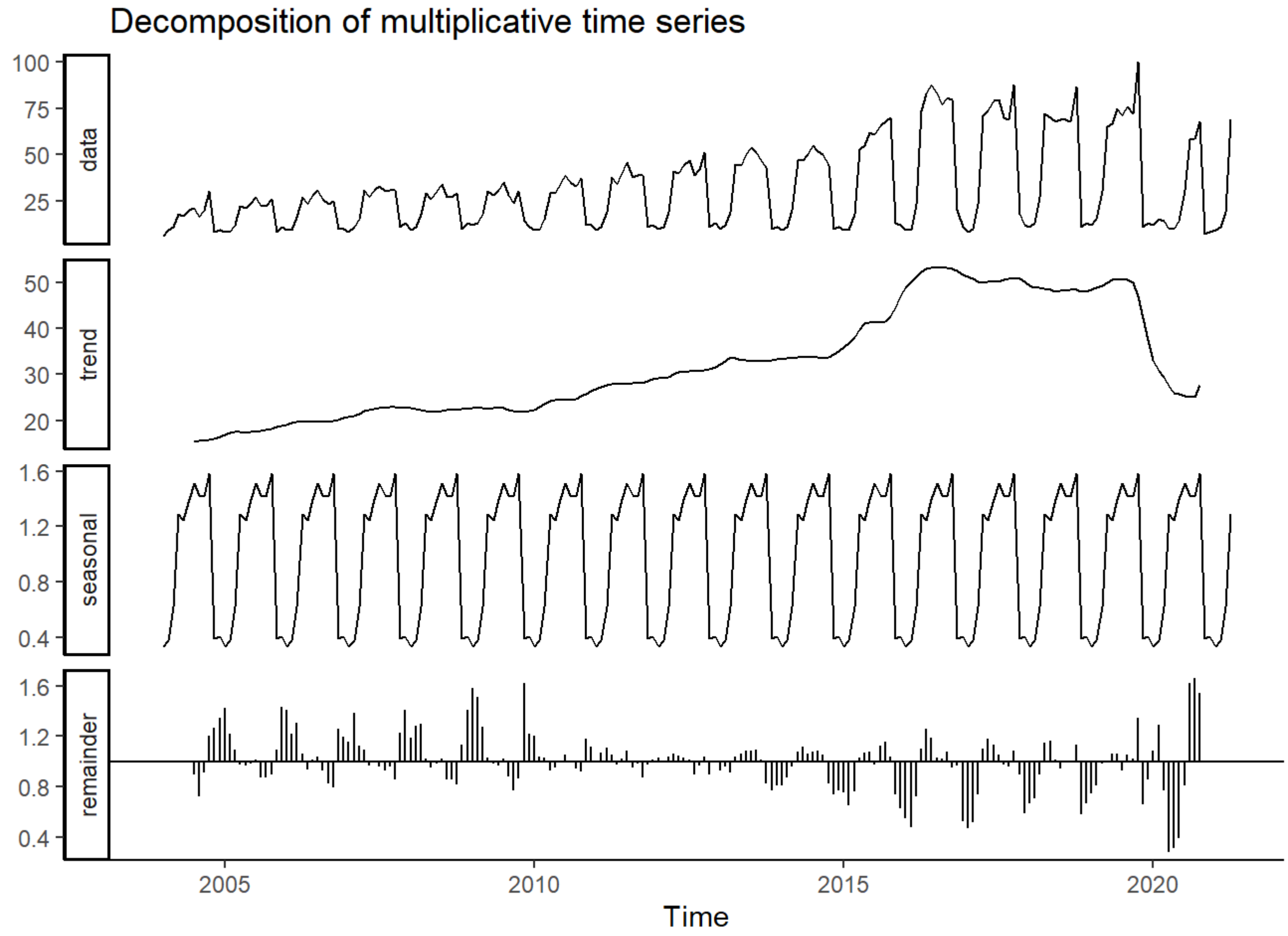
```
boxplot(mlbts~cycle(mlbts),xlab="Month", ylab = "mlb hits"
        ,main ="Monthly US mlb hits from 2004 to 2021")
```



decomposition

將原始資料、趨勢、季節性、殘差分別繪製成圖。

```
decomp_mlbts <- decompose(mlbts,"multiplicative")
autoplot(decomp_mlbts) + theme_classic()
```



Fitting Model method 1

Test stationality

以ADF test檢定平穩性，檢定結果顯著，此資料集為平穩序列。

```
adf.test(mlbts)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  mlbts
## Dickey-Fuller = -7.0113, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Fit arima

方法一將以auto.arima函數擬和模型。可得結果為ARIMA(1,0,1)(0,1,1)[12]。並可知AIC=1314.69。

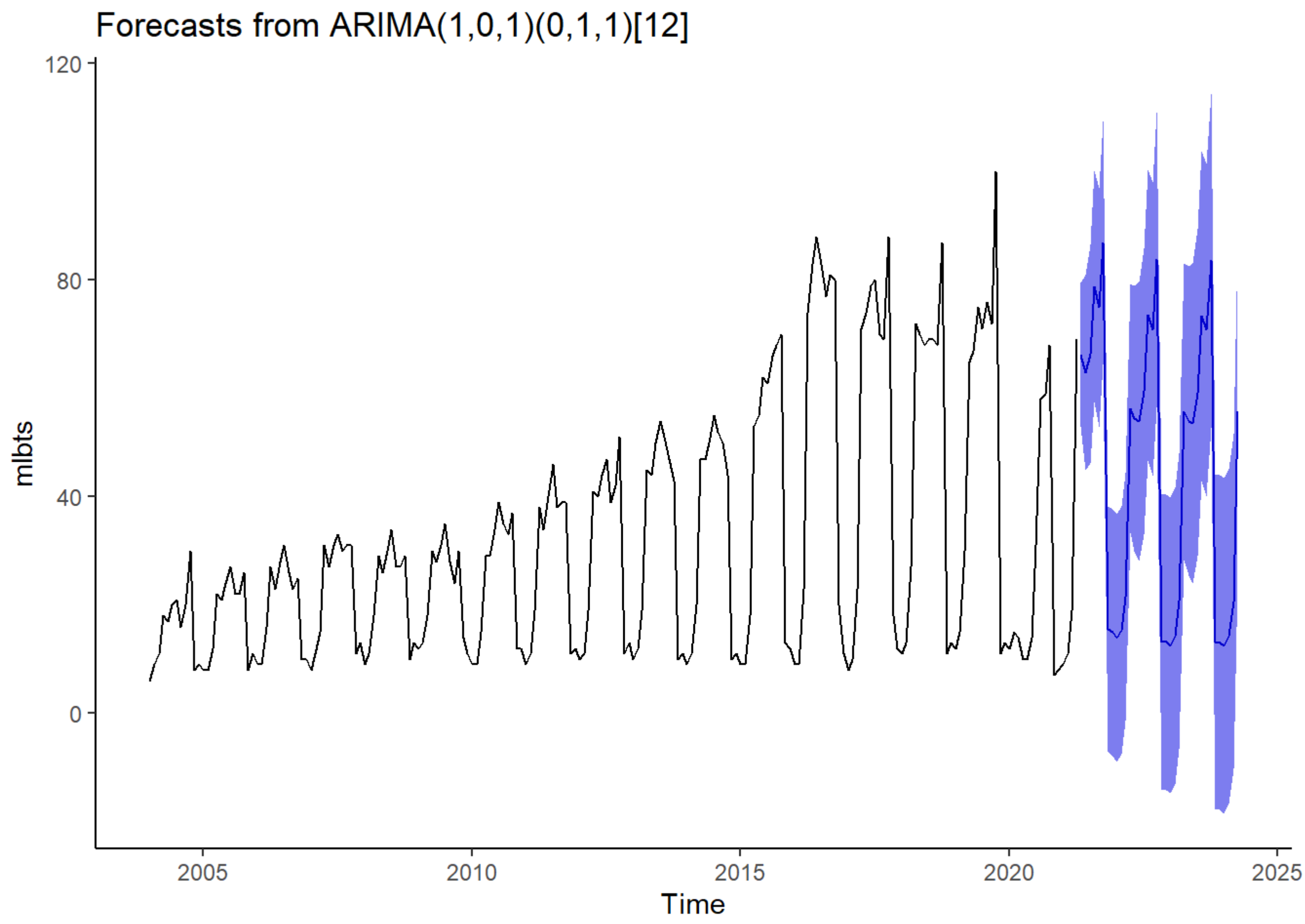
```
arima_mlbts <- auto.arima(mlbts)
arima_mlbts
```

```
## Series: mlbts
## ARIMA(1,0,1)(0,1,1)[12]
##
## Coefficients:
##          ar1     ma1     sma1
##       0.7650  0.1437  -0.3959
## s.e.  0.0584  0.0852   0.0669
##
## sigma^2 estimated as 45.99:  log likelihood=-653.34
## AIC=1314.69   AICc=1314.9   BIC=1327.8
```

Forcasting

繪製36步預測，並加上信賴區間。

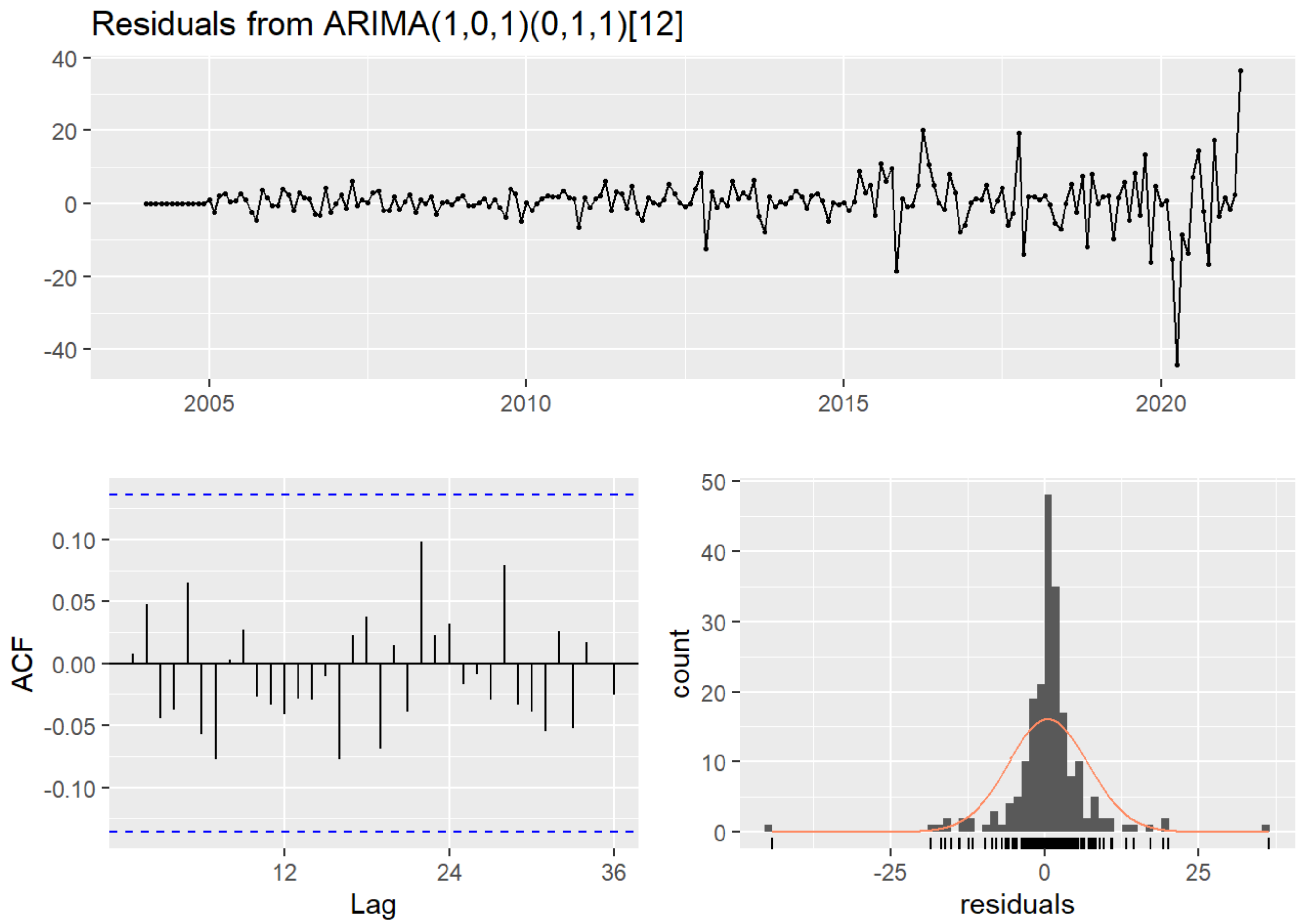
```
fore_mlbts <- forecast(arima_mlbts, level = c(95), h = 36)
autoplot(fore_mlbts) + theme_classic()
```



Residual

對殘差作圖分析，可從ACF圖中看出直接落於95%信賴區間中，顯示此模型對相關結構作很好的描述。並可從直方圖可看出殘差呈現常態分配。

```
checkresiduals(arima_mlbts)
```

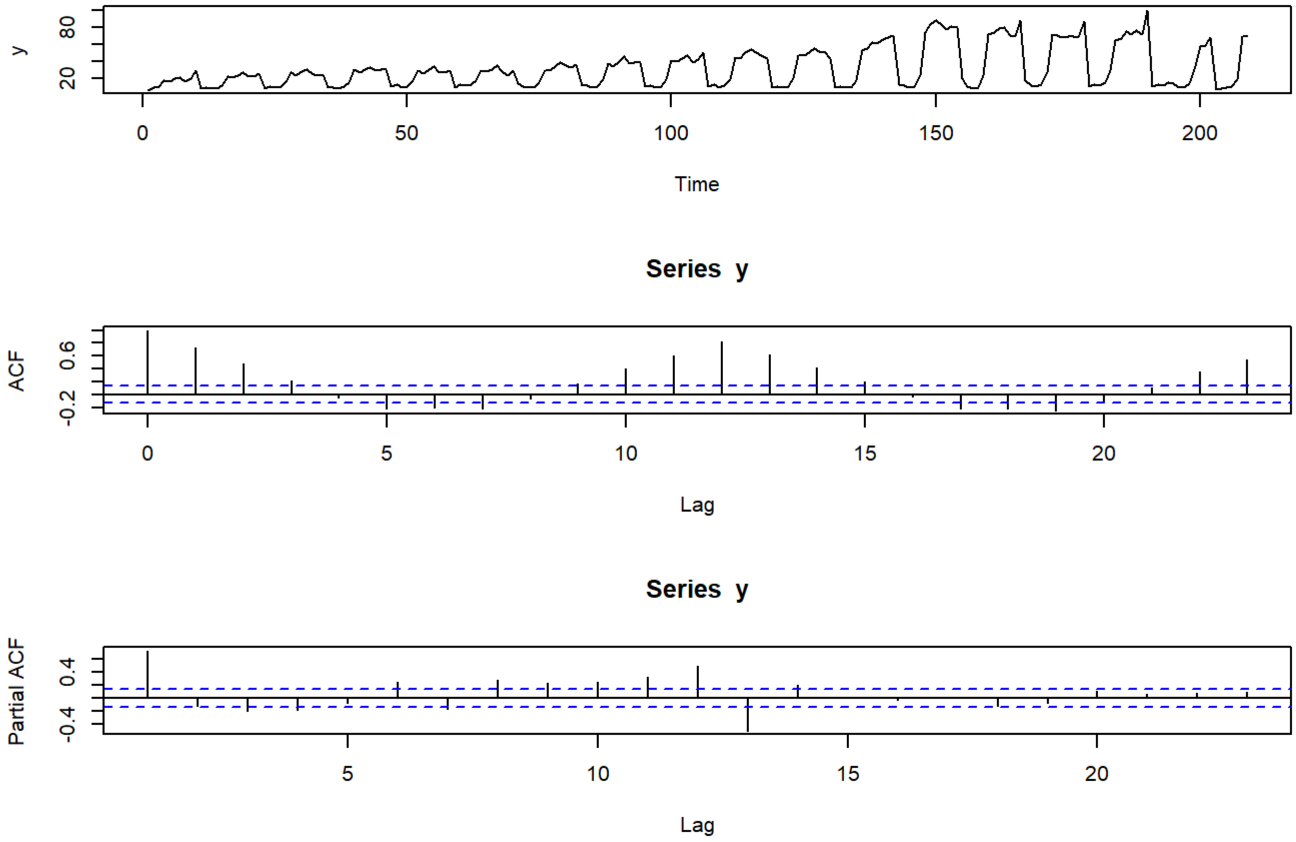


```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1)(0,1,1)[12]
## Q* = 11.417, df = 21, p-value = 0.954
##
## Model df: 3.   Total lags used: 24
```

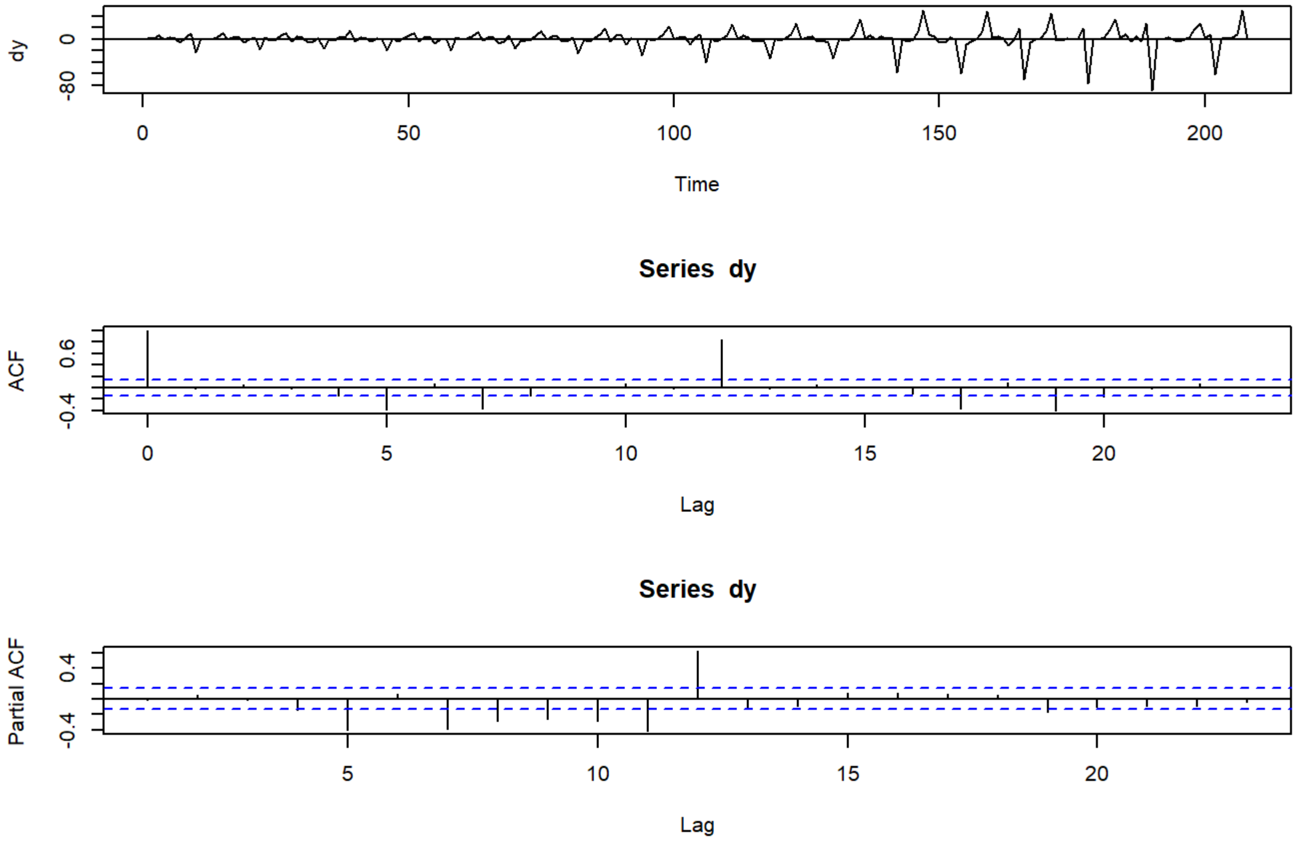
Fitting Model method 2 (Differencing)

方法二將直接觀察ACF、PACF圖形，並找出應擬合之模型。首先先將資料作一次差分，並可藉繪製出的ACF、PACF圖形看出，ACF從lag1處開始cutoff，對應至MA(1)模型。另外，對資料作季節性差分，可看出其ACF從lag1處開始cutoff，對應至MA(1)模型。故最後選擇SARIMA(0,1,1)*(0,1,1)，其中週期為12。並可從殘差之ACF圖看出似white noise。

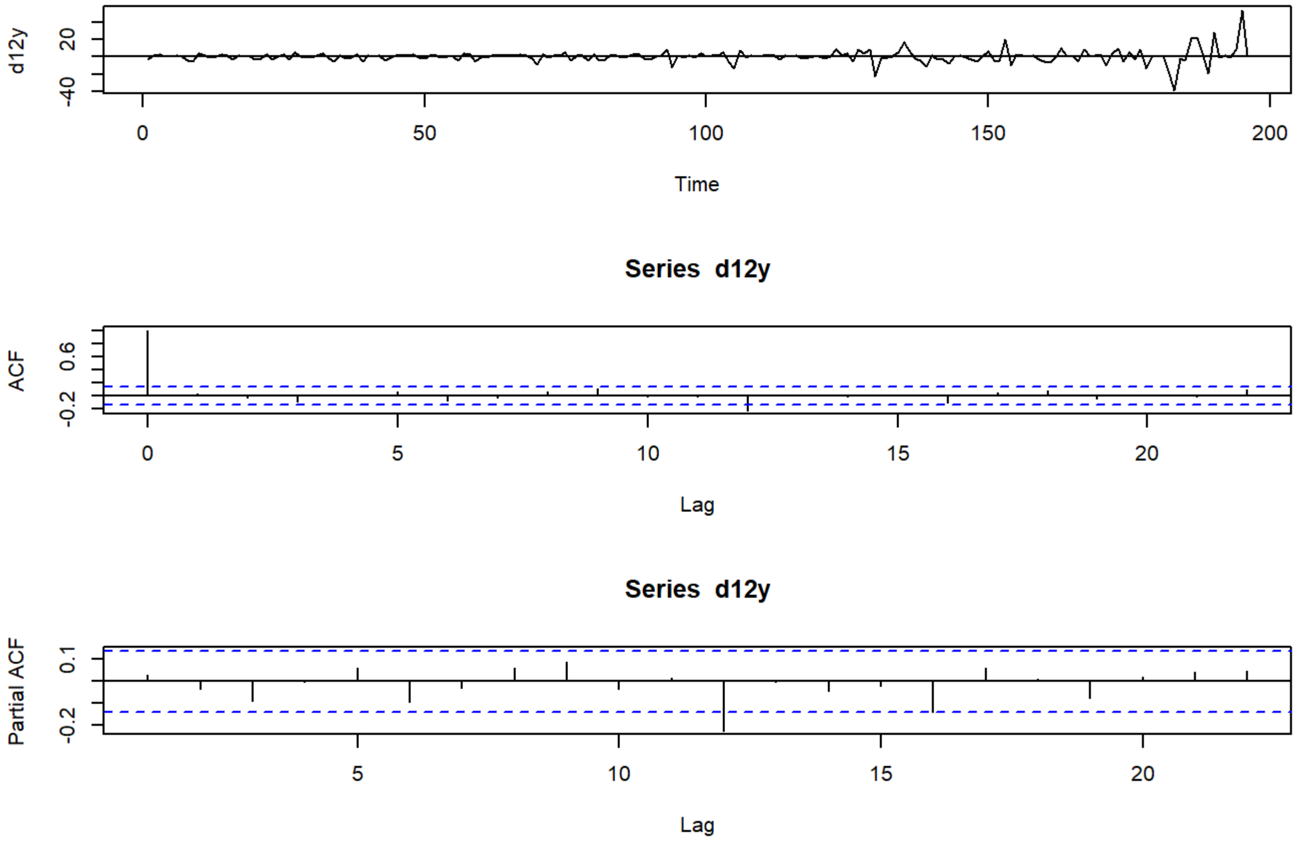
```
#original data
{par(mfrow=c(3,1))
y = a$interest_over_time[,2]
par(mfrow=c(3,1))
ts.plot(y)
acf(y)
pacf(y)}
```



```
#diff
dy<-diff(y)
{par(mfrow=c(3,1))
{ts.plot(dy)
 abline(h=mean(dy))
}
acf(dy)
pacf(dy)}
```

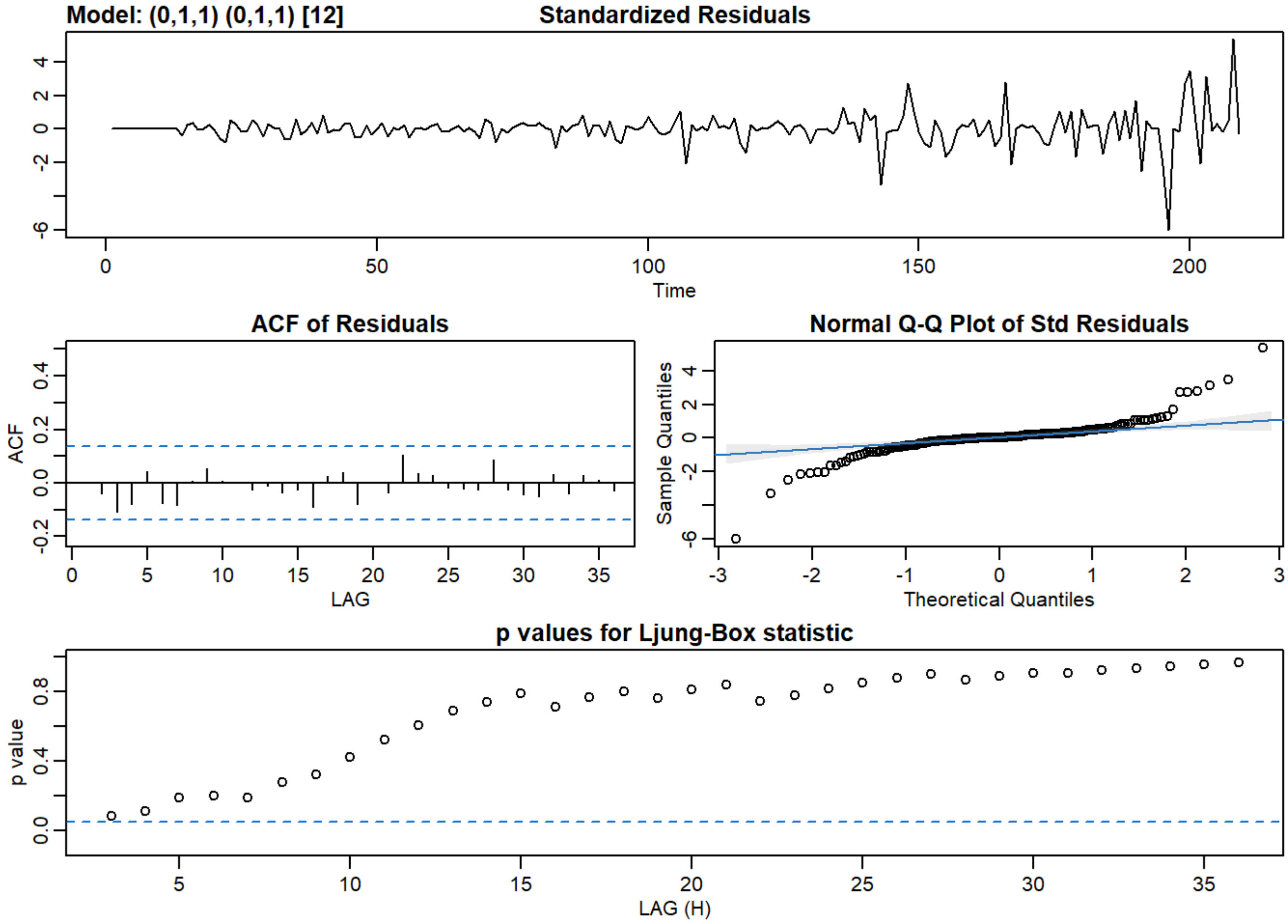


```
#seasonal diff
d12y<-diff(dy,12)
{par(mfrow=c(3,1))
{ts.plot(d12y)
abline(h=mean(d12y))
}
acf(d12y)
pacf(d12y)}
```



```
par(mfrow=c(1,1))
#choose model
sarima(y, 0,1,1,0,1,1,12)
```

```
## initial  value 2.011444
## iter   2 value 1.966717
## iter   3 value 1.962009
## iter   4 value 1.956331
## iter   5 value 1.955904
## iter   6 value 1.955888
## iter   7 value 1.955887
## iter   7 value 1.955887
## iter   7 value 1.955887
## final  value 1.955887
## converged
## initial  value 1.959941
## iter   2 value 1.959730
## iter   3 value 1.959730
## iter   3 value 1.959730
## iter   3 value 1.959730
## final  value 1.959730
## converged
```

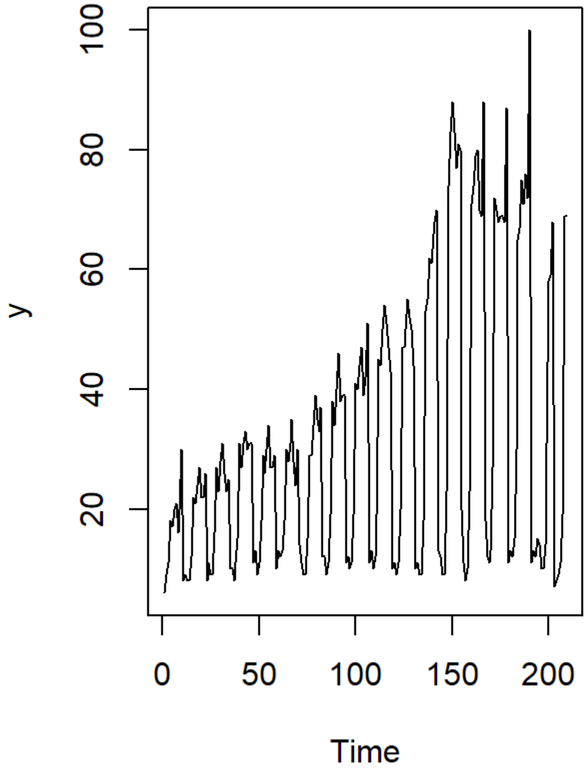

Model: (0,1,1) (0,1,1) [12]
Standardized Residuals
4
2
0
-2
-6
0
50
100
150
200
Time
ACF of Residuals
ACF
0.4
0.2
0.0
-0.2
0
5
10
15
20
25
30
35
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
4
2
0
-2
-6
-3
-2
-1
0
1
2
3
Theoretical Quantiles
p values for Ljung-Box statistic
p value
0.8
0.4
0.0
5
10
15
20
25
30
35
LAG (H)

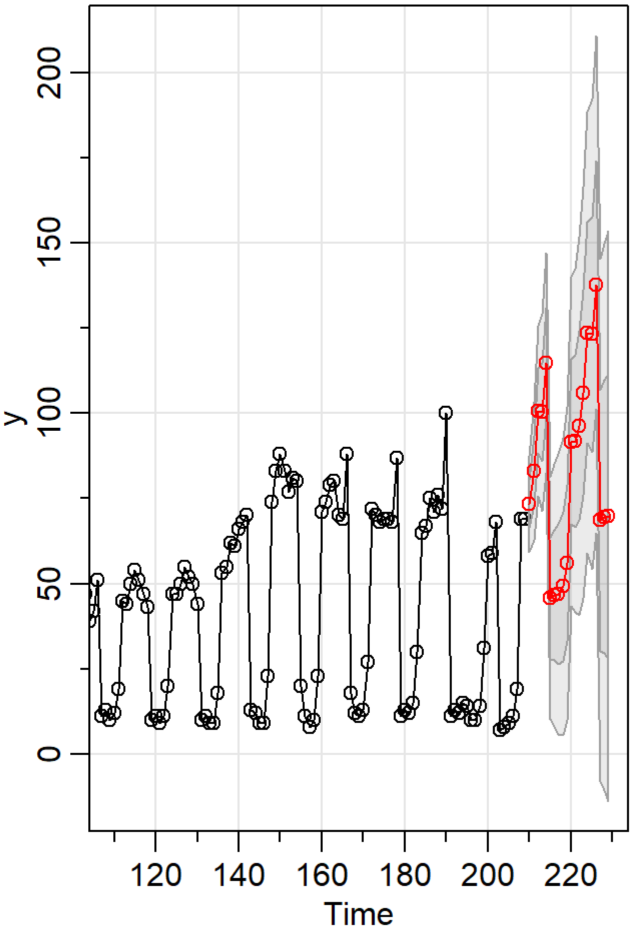
```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ma1     sma1
##       0.0313  -0.3775
## s.e.  0.0739   0.0714
##
## sigma^2 estimated as 49.9:  log likelihood = -662.22,  aic = 1330.44
##
## $degrees_of_freedom
## [1] 194
##
## $ttable
##      Estimate     SE t.value p.value
## ma1    0.0313 0.0739  0.4243  0.6718
## sma1  -0.3775 0.0714 -5.2873  0.0000
##
## $AIC
## [1] 6.427237
##
## $AICc
## [1] 6.427522
##
## $BIC
## [1] 6.474746
```

Forecast

同時繪製由方法一及方法二模型產生出的20步預測值。

```
par(mfrow=c(1,2))
{ts.plot(y)
sarima.for(y, 20, 0,1,1, 0,1,1, 12)}
```





```
## $pred
## Time Series:
## Start = 210
## End = 229
## Frequency = 1
##  [1]  73.41476  83.12410 100.70795 100.32272 114.81307  45.71099  46.66440
##  [8]  46.87129  49.18173  55.98280  91.72020  91.91464  96.34948 106.05882
## [15] 123.64266 123.25743 137.74778  68.64570  69.59911  69.80600
##
## $se
## Time Series:
## Start = 210
## End = 229
## Frequency = 1
##  [1]  7.064062 10.147853 12.492273 14.461495 16.192986 17.756428 19.192933
##  [8] 20.529166 21.783586 22.969601 24.097313 25.174560 27.753204 30.165562
## [15] 32.398796 34.487720 36.457149 38.325508 40.106924 41.812513
```

Efficiency

將資料切分成訓練集及驗證集，並計算以兩擬合後模型，套用至驗證集上的rmse。可得兩方法結果相近。

```
# splitting data into train and valid sets
trainy = z[1:168]
validy = z[168:length(y)]

# training model
modely = arima(trainy, order=c(1,0,0), season = list(order=c(0,1,1), period=12), method = 'ML')
model2y = arima(trainy, order=c(0,1,1), season = list(order=c(0,1,1), period=12), method = 'ML')

# model summary
summary(modely)
```

```
##
## Call:
## arima(x = trainy, order = c(1, 0, 0), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##          ar1     sma1
##       0.7142  -0.9999
## s.e.  0.0556   0.1760
##
## sigma^2 estimated as 91.21:  log likelihood = -589.55,  aic = 1185.1
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.07155714 9.202967 4.238517 -11.06154 33.12337 0.9040004
##                    ACF1
## Training set 0.04131448
```

```
summary(model2y)
```

```
##
## Call:
## arima(x = trainy, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##           ma1     sma1
##       -0.2478  -0.9997
## s.e.   0.1182   0.3289
##
## sigma^2 estimated as 104.2:  log likelihood = -595.83,  aic = 1197.66
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE     MASE
## Training set 0.02974055 9.804846 4.761052 -3.639438 38.30441 1.015448
##                    ACF1
## Training set 0.07195287
```

```
# forecasting
forecasty = predict(modely,42)
forecasty$pred
```

```
## Time Series:
## Start = 169
## End = 210
## Frequency = 1
##  [1] 18.902566 15.989099 12.172556 15.486086 12.922625  7.854478  8.741183
##  [8]  8.553450 11.841238 22.017023 16.282521 12.984394 14.606096 12.920471
## [15]  9.980878 13.920745 11.804625  7.055980  8.170879  8.146126 11.550320
## [22] 21.809243 16.134120 12.878404 14.530395 12.866404  9.942262 13.893164
## [29] 11.784927  7.041911  8.160831  8.138950 11.545194 21.805582 16.131505
## [36] 12.876536 14.529062 12.865451  9.941582 13.892678 11.784580  7.041663
```

```
forecast2y = predict(model2y,42)
forecast2y$pred
```

```
## Time Series:
## Start = 169
## End = 210
## Frequency = 1
##  [1] 18.50556 16.93413 14.07699 18.07698 16.00556 11.29128 12.43414 12.43414
##  [9] 15.86272 26.14842 20.50557 17.29131 18.96137 17.38994 14.53280 18.53279
## [17] 16.46136 11.74708 12.88994 12.88995 16.31852 26.60422 20.96138 17.74711
## [25] 19.41717 17.84574 14.98860 18.98859 16.91716 12.20289 13.34574 13.34575
## [33] 16.77432 27.06003 21.41718 18.20291 19.87298 18.30154 15.44440 19.44439
## [41] 17.37297 12.65869
```

```
# evaluation
rmse(validy, forecasty$pred)
```

```
## [1] 15.74137
```

```
rmse(validy, forecast2y$pred)
```

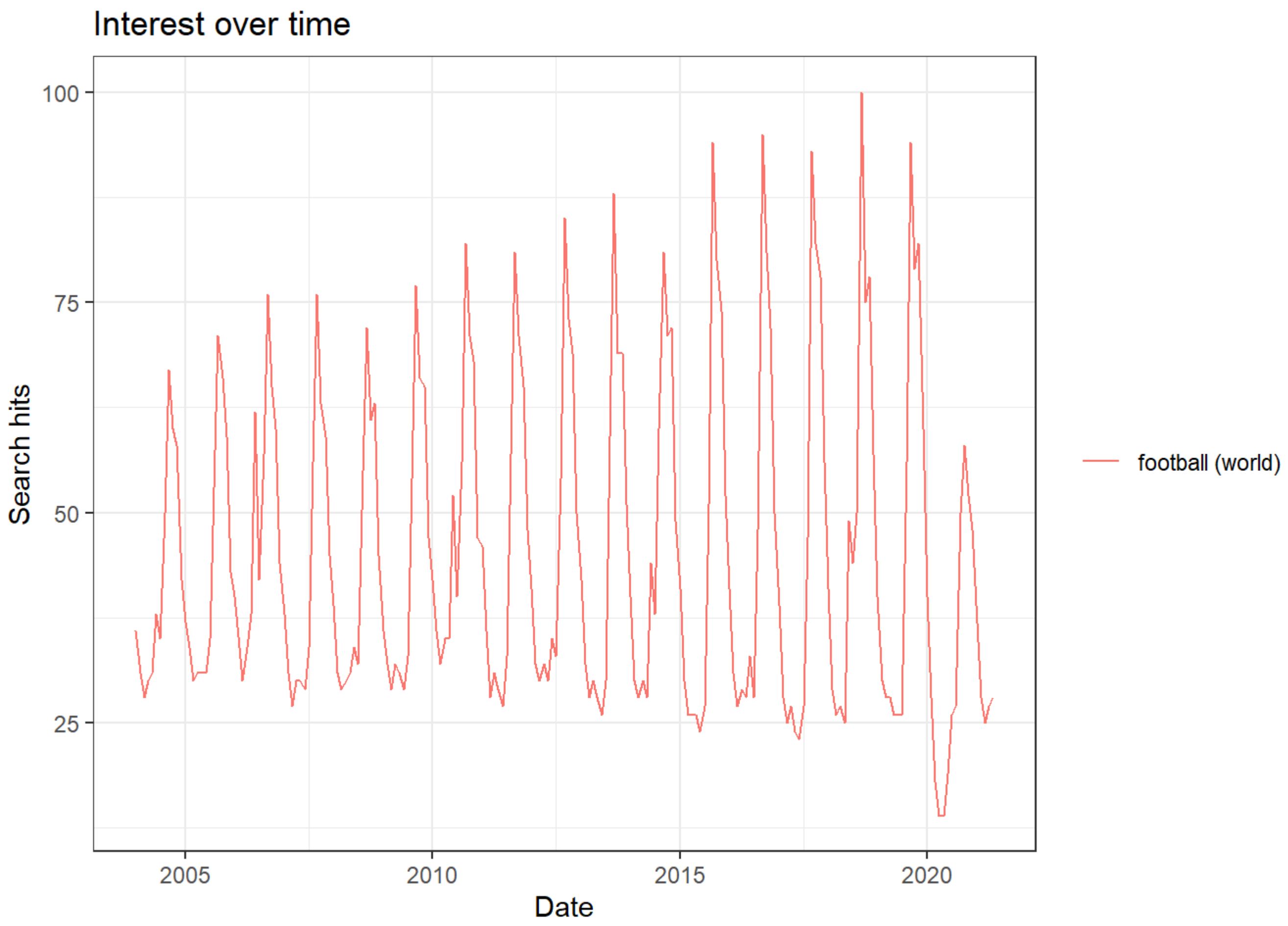
```
## [1] 14.50196
```

Dataset Football

Basic information

此為關鍵字為football之資料集，並針對其作線圖及直方圖。

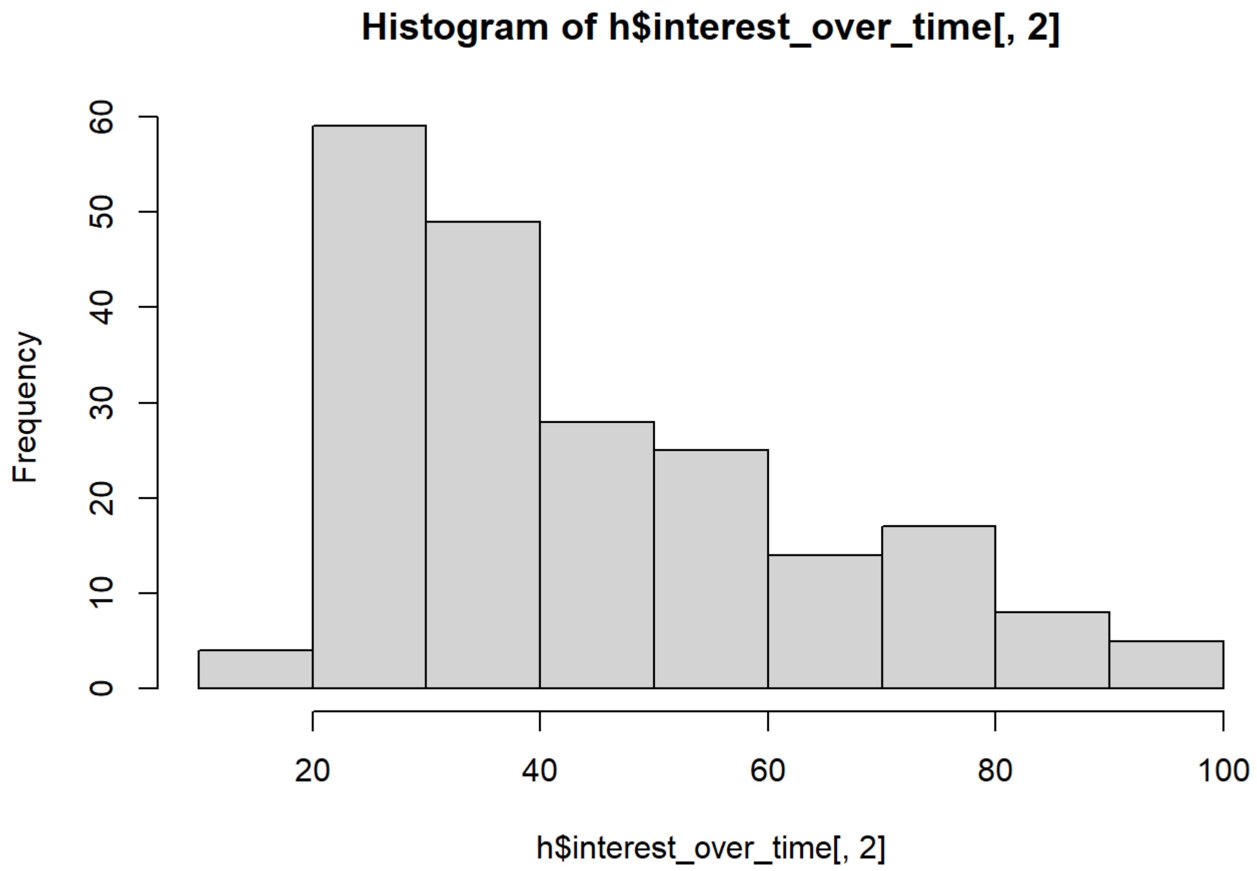
```
h = gtrends("football", time="all")
plot(h)
```



```
names(h)
```

```
## [1] "interest_over_time"  "interest_by_country" "interest_by_region"
## [4] "interest_by_dma"     "interest_by_city"    "related_topics"
## [7] "related_queries"
```

```
hist(h$interest_over_time[,2], 10)
```



```
head(h$related_topics)
```

```
##   subject related_topics                            value  keyword category
## 1     100            top                         Football football        0
## 2      91            top                American football football        0
## 3      13            top                        BBC Sport football        0
## 4      11            top                     BBC Scotland football        0
## 5      11            top British Broadcasting Corporation football        0
## 6      10            top                 College Football football        0
```

EDA

```
us_foot<-h$interest_over_time[,1:2]
str(us_foot)
```

```
## 'data.frame':    209 obs. of  2 variables:
##  $ date: POSIXct, format: "2004-01-01" "2004-02-01" ...
##  $ hits: int  36 31 28 30 31 38 35 50 67 60 ...
```

```
us_foot[,1]<-as.factor(us_foot[,1])
attach(us_foot)
```

```
## The following objects are masked from us_mlb:
##
##     date, hits
```

```
## The following objects are masked from us_flu:
##
##     date, hits
```

```
footts<-ts(hits,c(2004,1),c(2021,4),12)
str(footts)
```

```
##  Time-Series [1:208] from 2004 to 2021: 36 31 28 30 31 38 35 50 67 60 ...
```

```
footts
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2004  36  31  28  30  31  38  35  50  67  60  58  42
## 2005  37  34  30  31  31  31  35  53  71  66  59  43
## 2006  40  34  30  34  38  62  42  57  76  65  60  44
## 2007  38  31  27  30  30  29  34  56  76  63  59  45
## 2008  39  31  29  30  31  34  32  55  72  61  63  45
## 2009  36  32  29  32  31  29  33  54  77  66  65  47
## 2010  42  36  32  35  35  52  40  56  82  71  68  47
## 2011  46  36  28  31  29  27  33  57  81  71  65  48
## 2012  41  32  30  32  30  35  33  54  85  73  69  50
## 2013  42  32  28  30  28  26  30  59  88  69  69  51
## 2014  40  30  28  30  28  44  38  60  81  71  72  49
## 2015  41  30  26  26  26  24  27  51  94  80  74  52
## 2016  41  31  27  29  28  33  28  48  95  81  72  50
## 2017  39  28  25  27  24  23  27  47  93  82  78  56
## 2018  41  29  26  27  25  49  44  50 100  75  78  57
## 2019  40  30  28  28  26  26  26  56  94  79  82  60
## 2020  40  28  18  14  14  19  26  27  50  58  52  48
## 2021  39  28  25  27
```

```
frequency(footts)
```

```
## [1] 12
```

```
cycle(footts)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2004   1   2   3   4   5   6   7   8   9  10  11  12
## 2005   1   2   3   4   5   6   7   8   9  10  11  12
## 2006   1   2   3   4   5   6   7   8   9  10  11  12
## 2007   1   2   3   4   5   6   7   8   9  10  11  12
## 2008   1   2   3   4   5   6   7   8   9  10  11  12
## 2009   1   2   3   4   5   6   7   8   9  10  11  12
## 2010   1   2   3   4   5   6   7   8   9  10  11  12
## 2011   1   2   3   4   5   6   7   8   9  10  11  12
## 2012   1   2   3   4   5   6   7   8   9  10  11  12
## 2013   1   2   3   4   5   6   7   8   9  10  11  12
## 2014   1   2   3   4   5   6   7   8   9  10  11  12
## 2015   1   2   3   4   5   6   7   8   9  10  11  12
## 2016   1   2   3   4   5   6   7   8   9  10  11  12
## 2017   1   2   3   4   5   6   7   8   9  10  11  12
## 2018   1   2   3   4   5   6   7   8   9  10  11  12
## 2019   1   2   3   4   5   6   7   8   9  10  11  12
## 2020   1   2   3   4   5   6   7   8   9  10  11  12
## 2021   1   2   3   4
```

```
summary(footts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   14.00   30.00   39.00   45.03   58.00  100.00
```

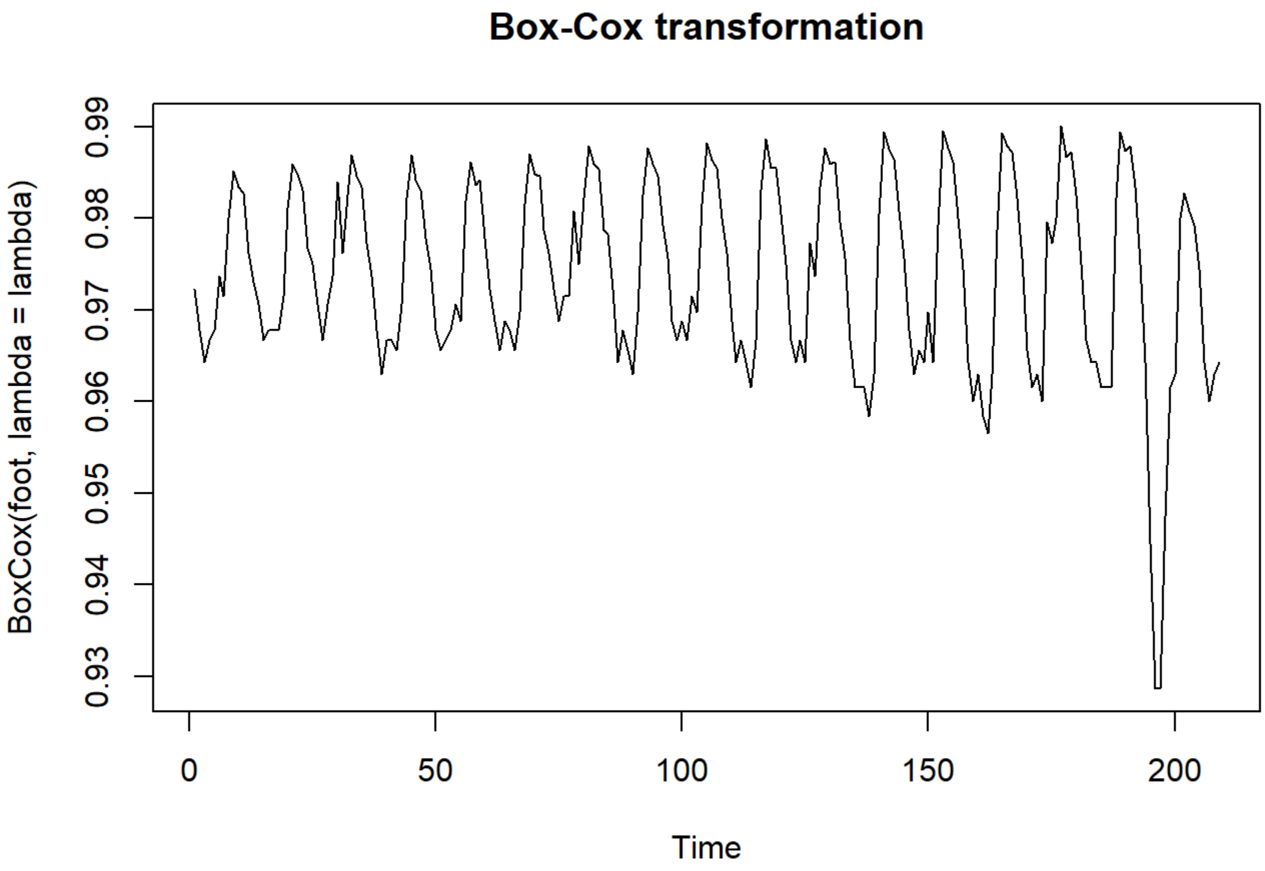
Box-cox

利用Box-Cox transformation，使轉換後的資料變異數齊一，更似常態分佈。 其中，計算出的lambda值為-0.9999242，並將轉換後的資料繪製成圖。

```
par(mfrow=c(1,1))
foot<-ts(h$interest_over_time[,2])
lambda <- BoxCox.lambda(log(foot))
print(lambda)
```

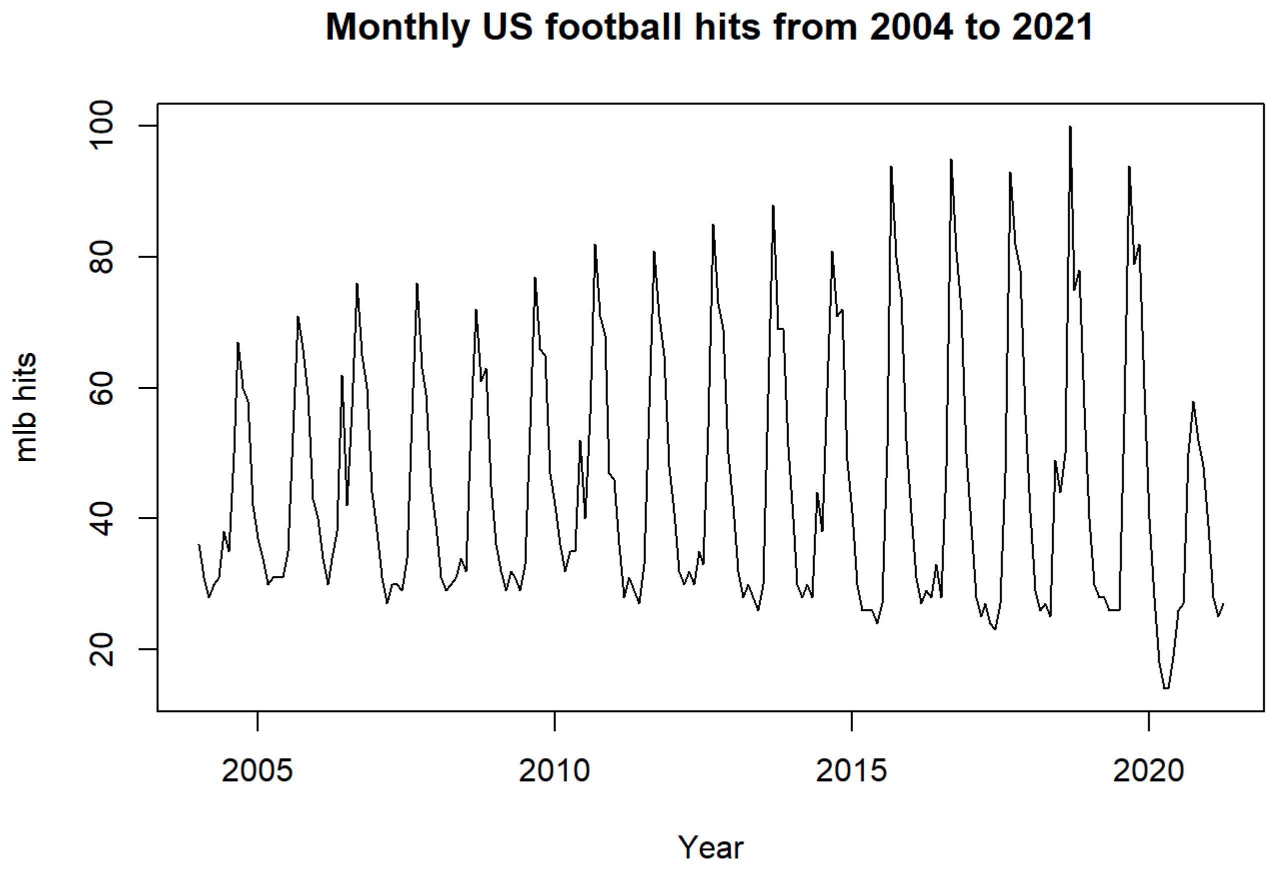
```
## [1] -0.9999242
```

```
plot.ts(BoxCox(foot, lambda = lambda), main='Box-Cox transformation')
```



TS-plot

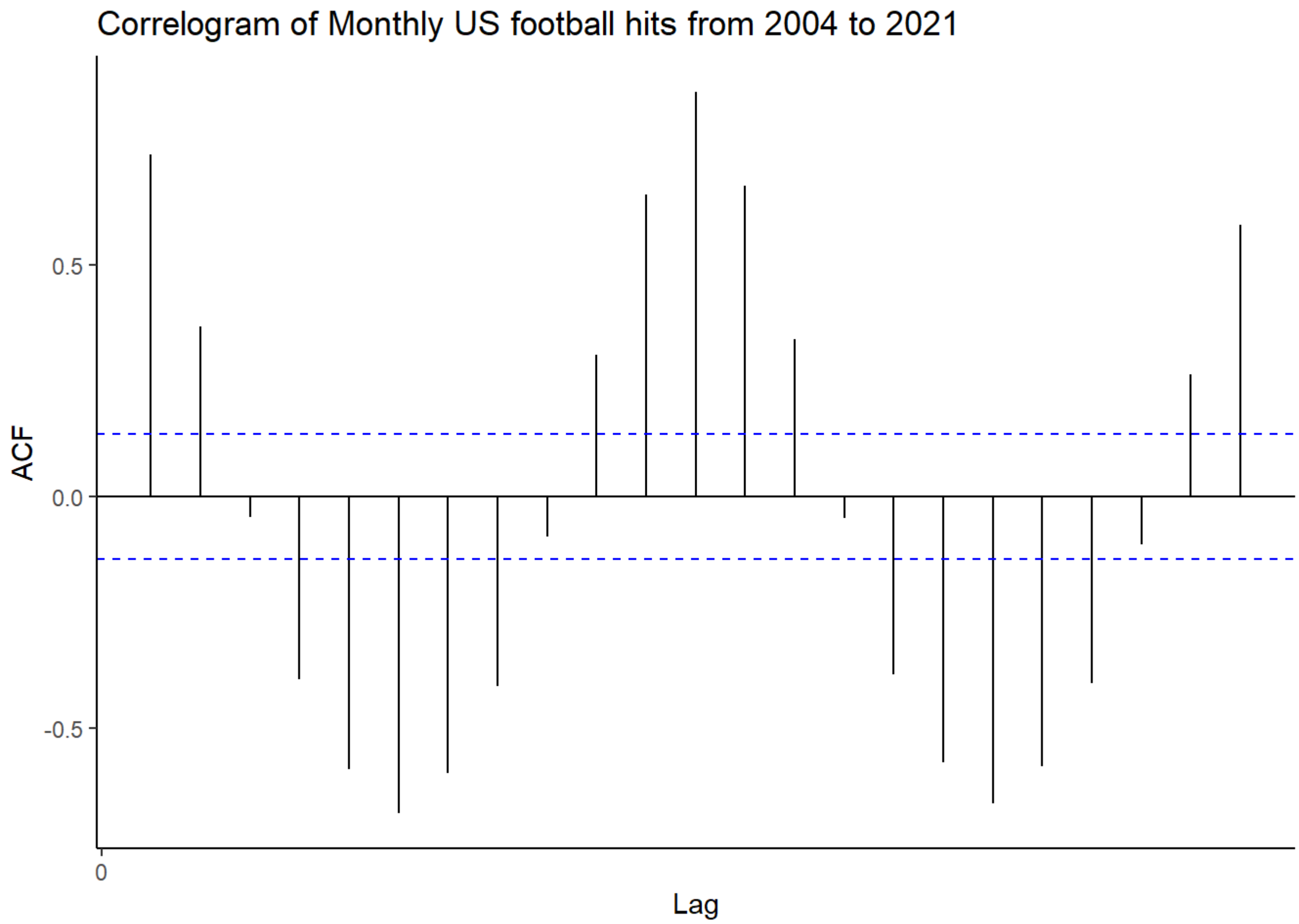
```
plot(footts,xlab="Year", ylab = "mlb hits",
     main="Monthly US football hits from 2004 to 2021")
```



ACF of footts

繪製資料之ACF圖

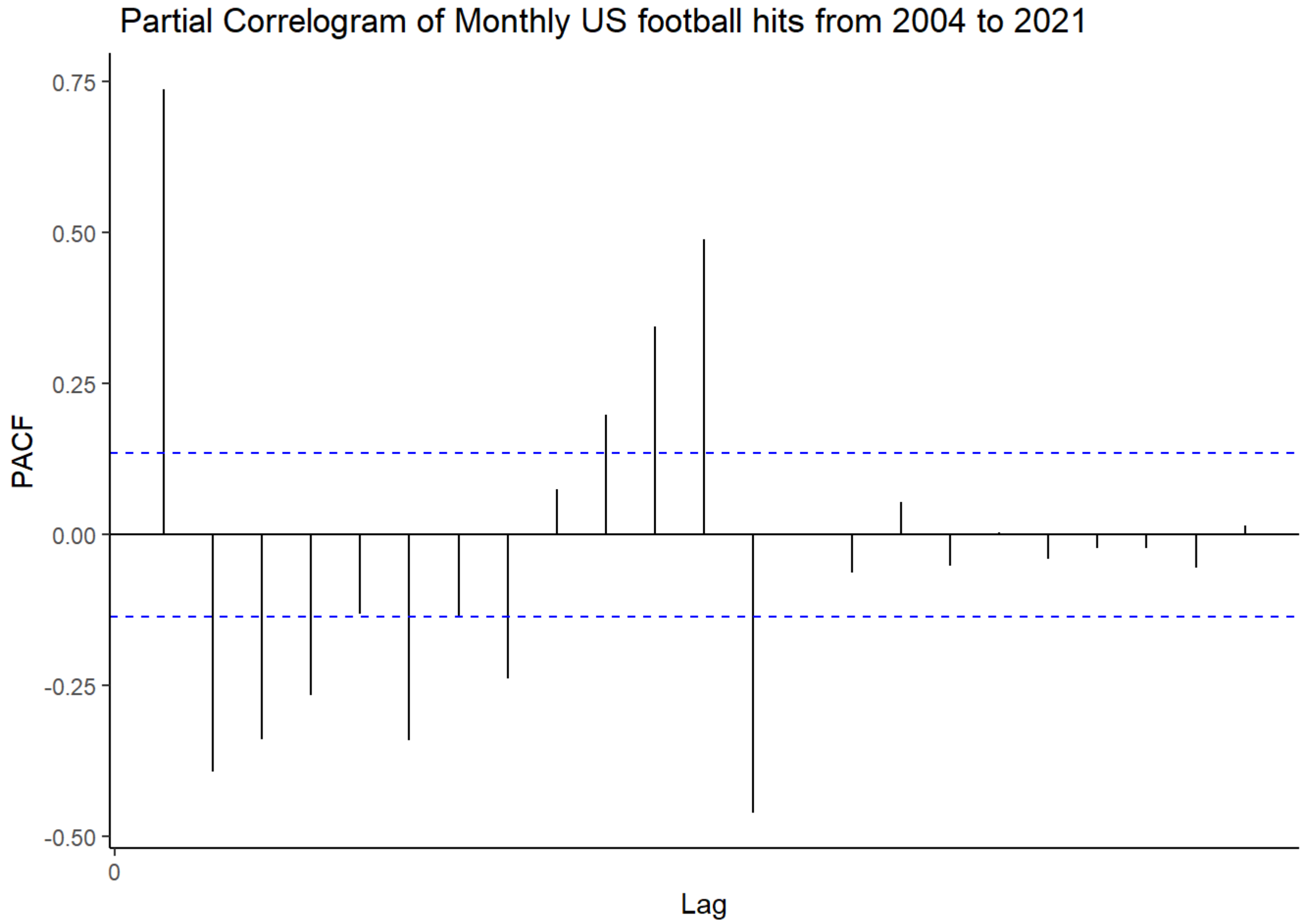
```
autoplot(acf(footts,plot=FALSE))+
  labs(title="Correlogram of Monthly US football hits from 2004 to 2021") + theme_classic()
```



PACF of footts

繪製資料之PACF圖

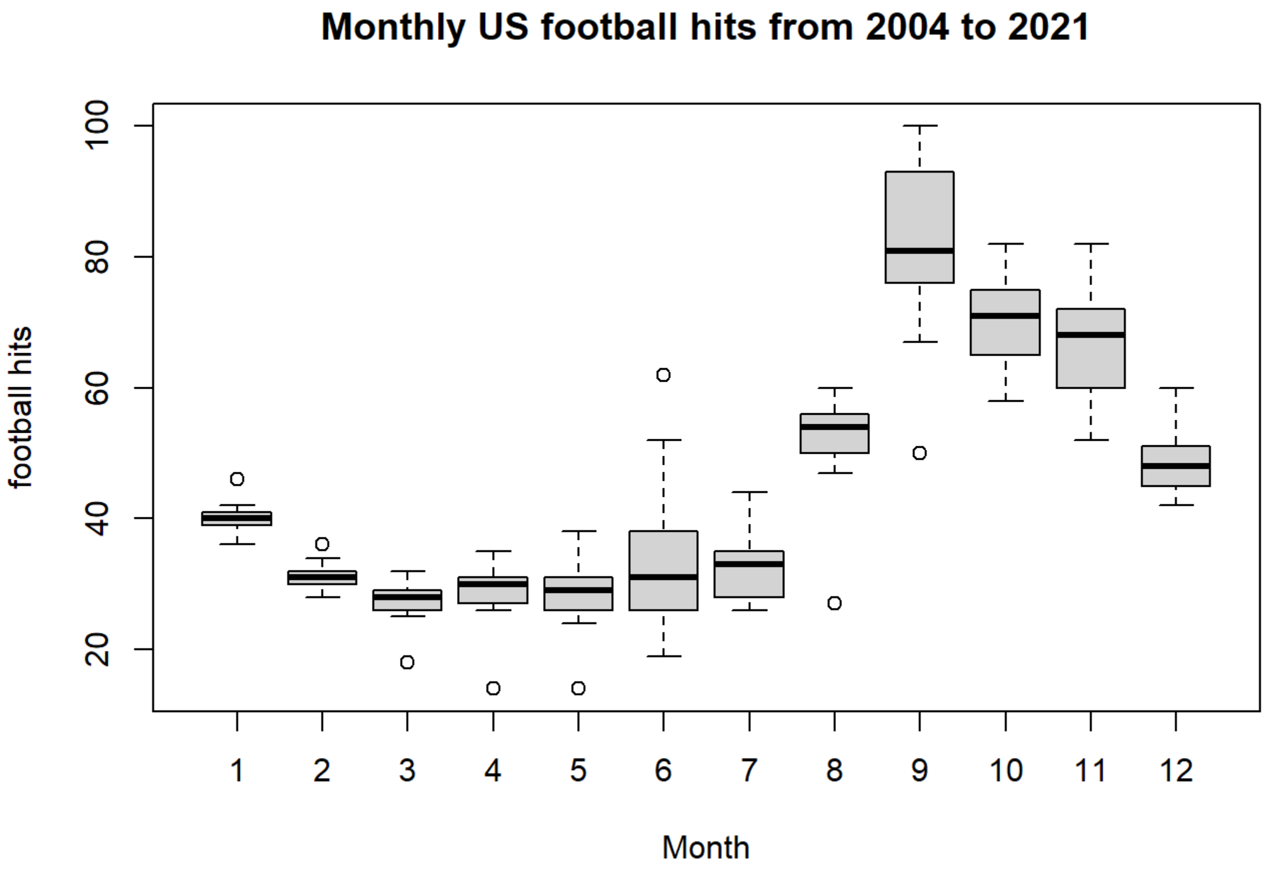
```
autoplot(pacf(footts,plot=FALSE))+
  labs(title=" Partial Correlogram of Monthly US football hits from 2004 to 2021") + theme_classic()
```



Boxplot

繪製資料之盒鬚圖，可看出2004~2021平均而言，關鍵字搜尋次數於9月份到達最高峰。

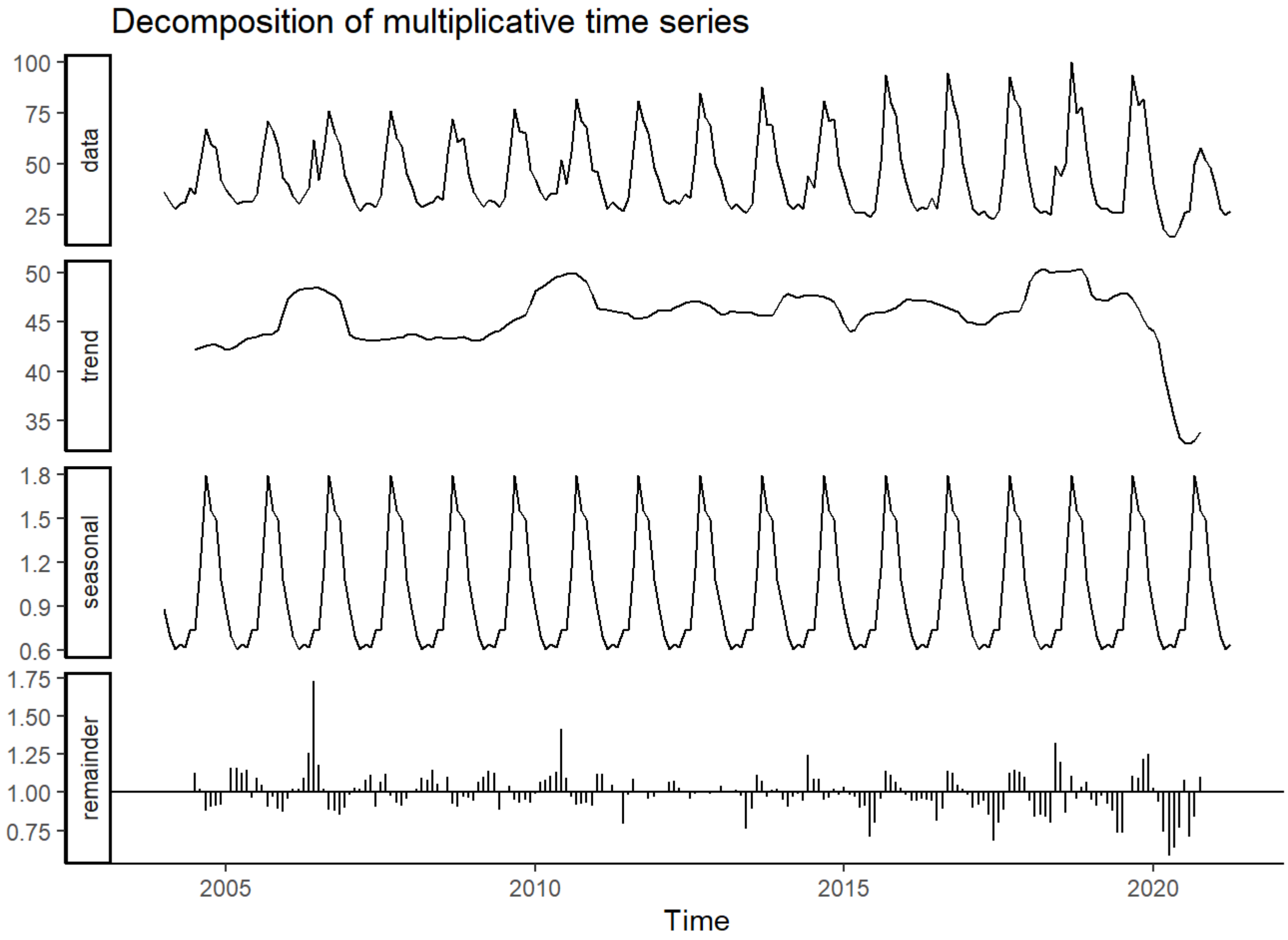
```
boxplot(footts~cycle(footts),xlab="Month", ylab = "football hits"
        ,main ="Monthly US football hits from 2004 to 2021")
```



decomposition

將原始資料、趨勢、季節性、殘差分別繪製成圖。

```
decomp_footts <- decompose(footts,"multiplicative")
autoplot(decomp_footts) + theme_classic()
```



Fitting Model method 1

Test stationality

以ADF test檢定平穩性，檢定結果顯著，此資料集為平穩序列。

```
adf.test(footts)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  footts
## Dickey-Fuller = -11.386, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Fit arima

方法一將以auto.arima函數擬和模型。可得結果為ARIMA(1,0,0)(0,1,2)[12] 。並可知AIC=1231.24。

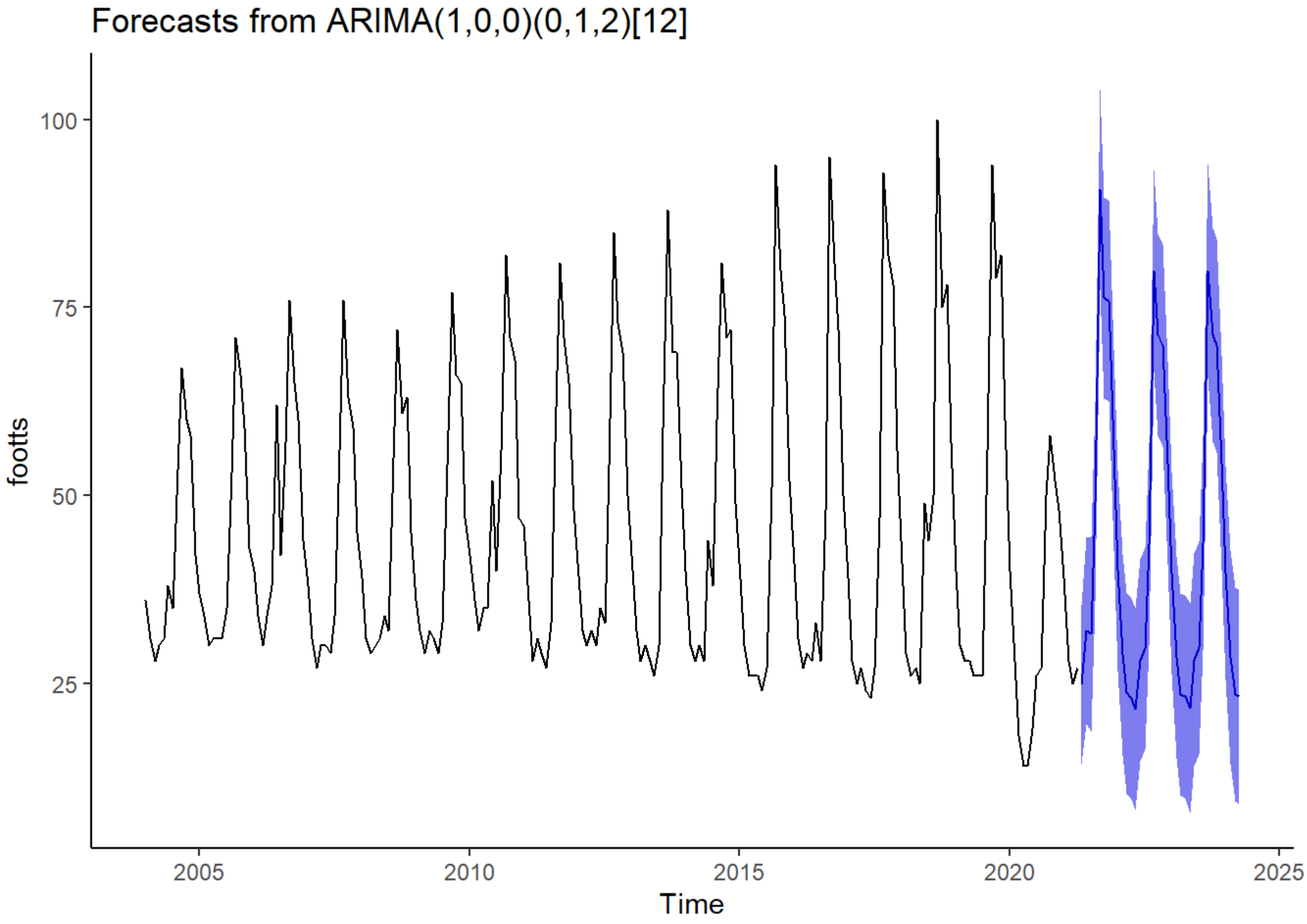
```
arima_footts <- auto.arima(footts)
arima_footts
```

```
## Series: footts
## ARIMA(1,0,0)(0,1,2)[12]
##
## Coefficients:
##          ar1     sma1    sma2
##       0.6157  -0.8912  0.2379
## s.e.  0.0560   0.0922  0.0938
##
## sigma^2 estimated as 28.91:  log likelihood=-611.62
## AIC=1231.24   AICc=1231.45   BIC=1244.36
```

Forcasting

繪製36步預測，並加上信賴區間。

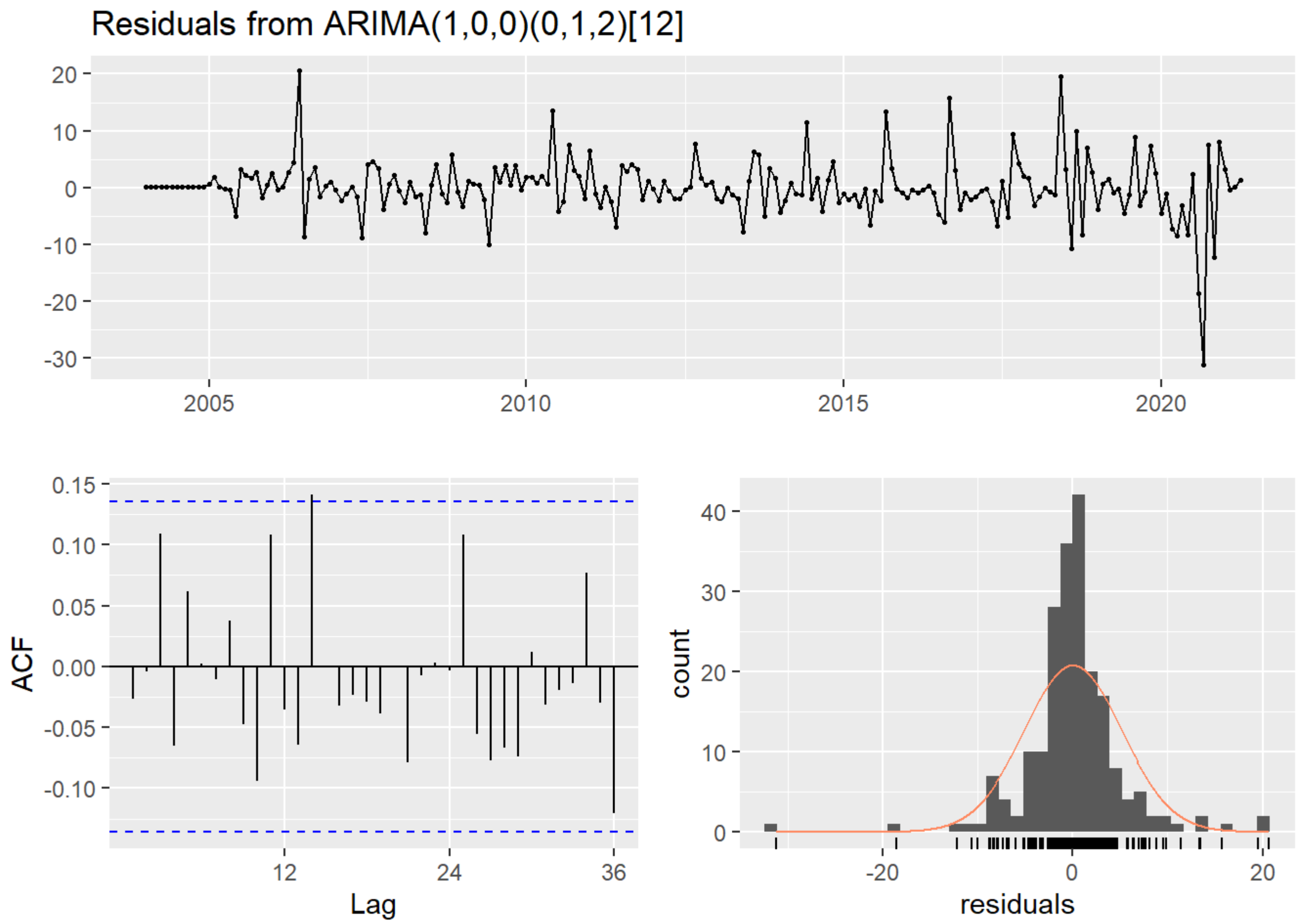
```
fore_footts <- forecast(arima_footts, level = c(95), h = 36)
autoplot(fore_footts) + theme_classic()
```



Residual

對殘差作圖分析，可從ACF圖中看出直接落於95%信賴區間中，顯示此模型對相關結構作很好的描述。並可從直方圖可看出殘差呈現常態分配。

```
checkresiduals(arima_footts)
```

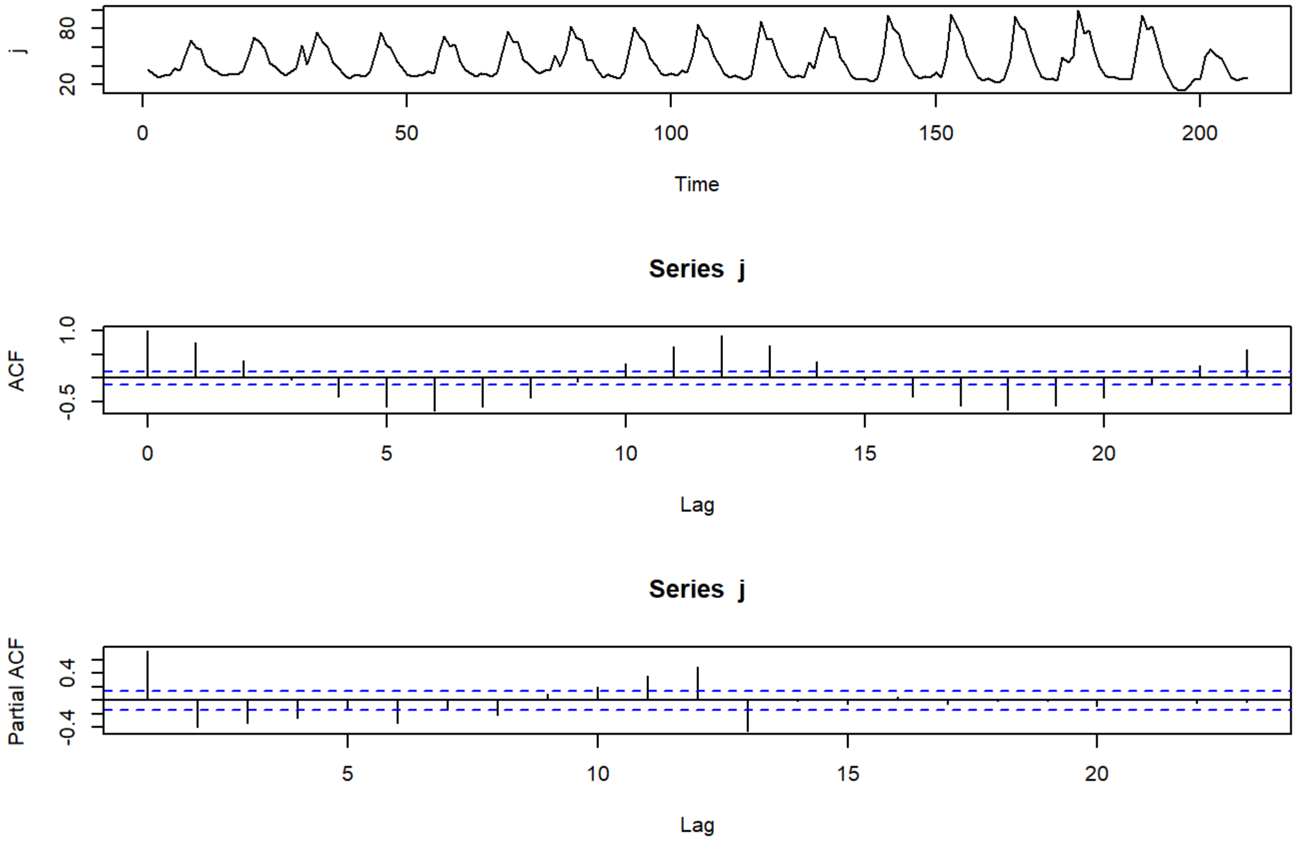


```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(0,1,2)[12]
## Q* = 17.83, df = 21, p-value = 0.6597
##
## Model df: 3.   Total lags used: 24
```

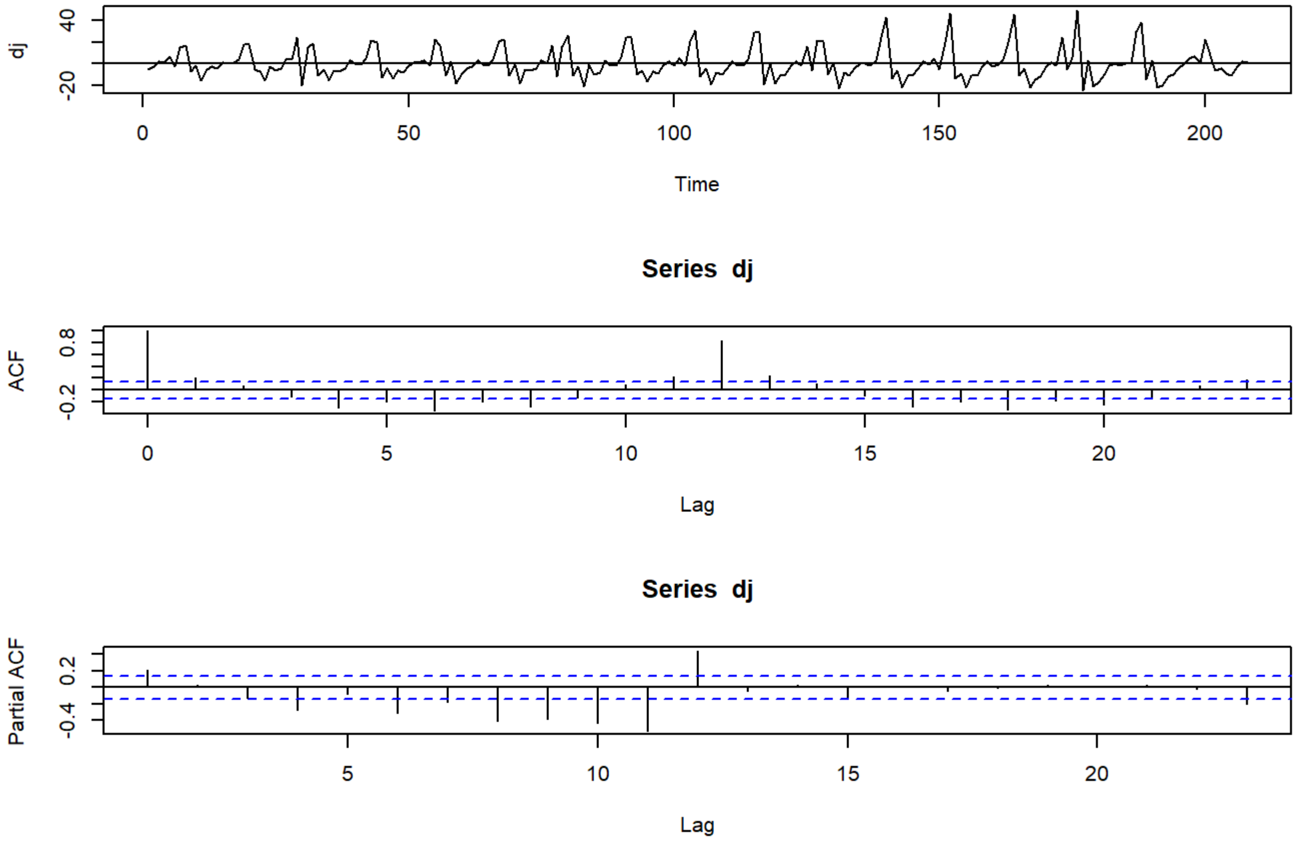
Fitting Model method 2 (Differencing)

方法二將直接觀察ACF、PACF圖形，並找出應擬合之模型。首先先將資料作一次差分，並可藉繪製出的ACF、PACF圖形看出，ACF從lag1處開始tailoff，對應至AR(1)模型。且PACF從lag1開始cutoff，對應至MA(1)模型。另外，對資料作季節性差分，可看出其ACF從lag1處開始tailoff，對應至AR(1)模型。且PACF從lag1開始cutoff，對應至MA(1)模型。故最後選擇SARIMA(1,1,1)*(1,1,1)，其中週期為12。並可從殘差之ACF圖看出似white noise。

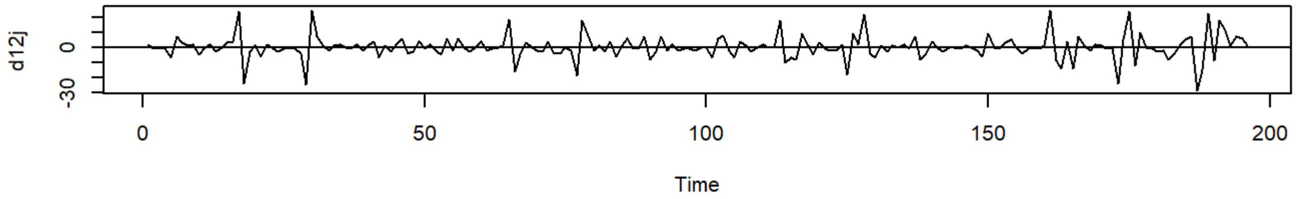
```
#original data
{par(mfrow=c(3,1))
  j = h$interest_over_time[,2]
  par(mfrow=c(3,1))
  ts.plot(j)
  acf(j)
  pacf(j)}
```

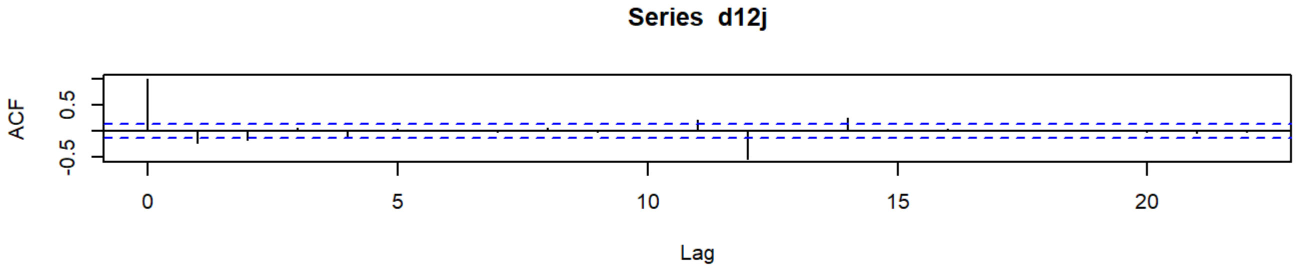


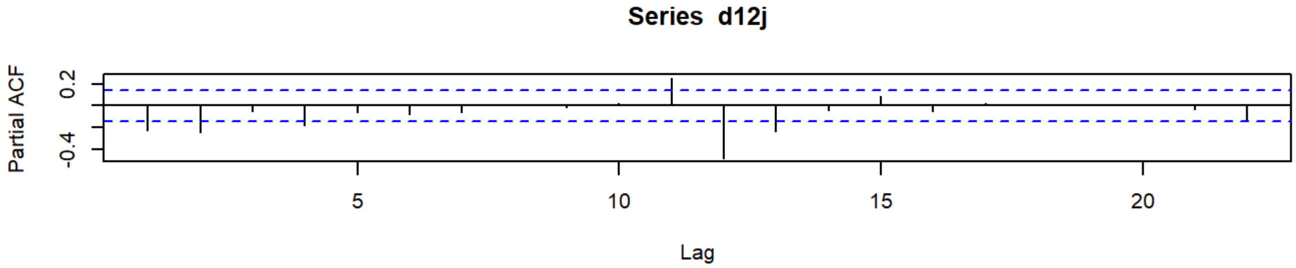
```
#diff
dj<-diff(j)
{par(mfrow=c(3,1))
  {ts.plot(dj)
    abline(h=mean(dj))
}
  acf(dj)
  pacf(dj)}
```



```
#seasonal diff
d12j<-diff(dj,12)
{par(mfrow=c(3,1))
  {ts.plot(d12j)
    abline(h=mean(d12j))
}
  acf(d12j)
  pacf(d12j)}
```

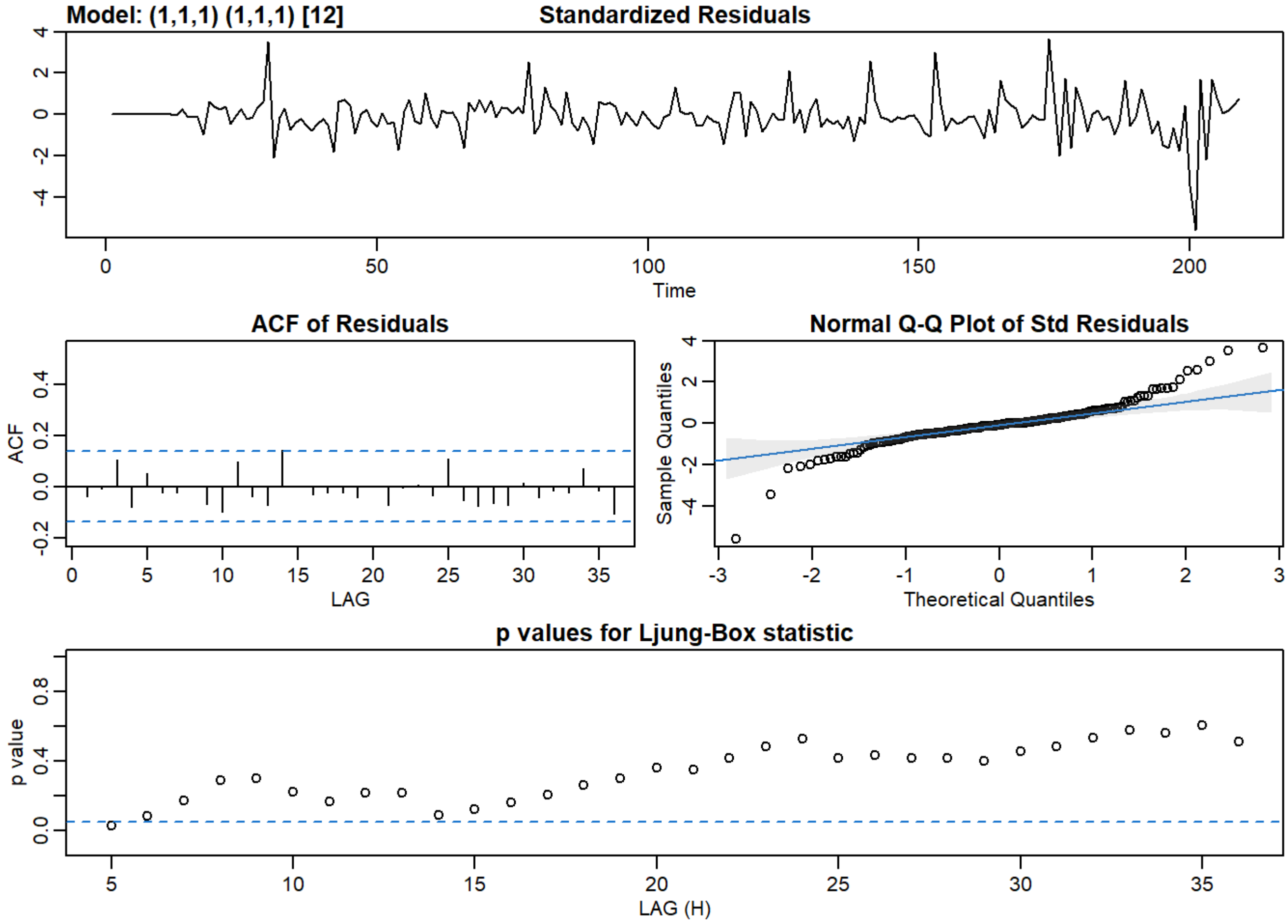






```
par(mfrow=c(1,1))
#choose model
sarima(j, 1,1,1,1,1,1,12)
```

```
## initial  value 2.083439
## iter   2 value 1.797376
## iter   3 value 1.783406
## iter   4 value 1.780867
## iter   5 value 1.777070
## iter   6 value 1.770846
## iter   7 value 1.740392
## iter   8 value 1.730666
## iter   9 value 1.730615
## iter  10 value 1.728436
## iter  11 value 1.727951
## iter  12 value 1.727813
## iter  13 value 1.727803
## iter  14 value 1.727802
## iter  14 value 1.727802
## iter  14 value 1.727802
## final  value 1.727802
## converged
## initial  value 1.719310
## iter   2 value 1.717494
## iter   3 value 1.717439
## iter   4 value 1.715933
## iter   5 value 1.715317
## iter   6 value 1.715078
## iter   7 value 1.714853
## iter   8 value 1.714817
## iter   9 value 1.714811
## iter  10 value 1.714810
## iter  10 value 1.714810
## final  value 1.714810
## converged
```

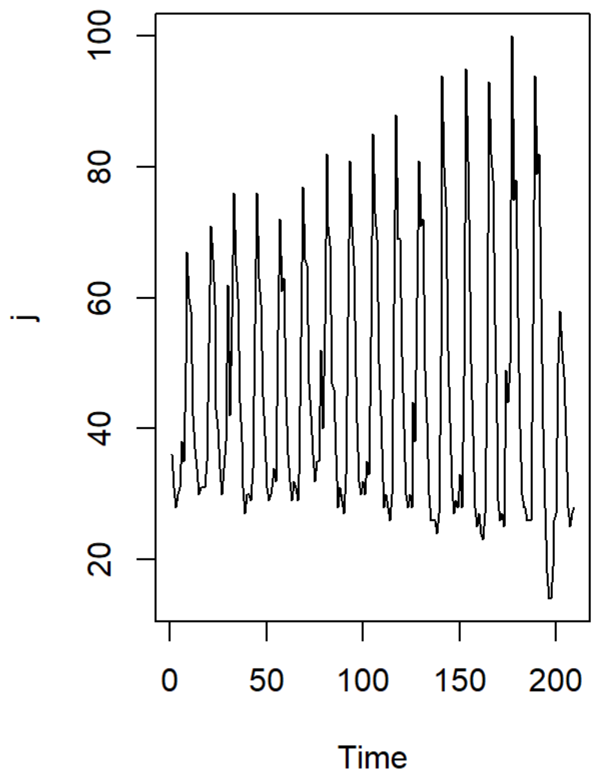

Model: (1,1,1) (1,1,1) [12]
Standardized Residuals
4
2
0
-2
-4
0
50
100
150
200
Time
ACF of Residuals
ACF
0.4
0.2
0.0
-0.2
0
5
10
15
20
25
30
35
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
4
2
0
-2
-4
-3
-2
-1
0
1
2
3
Theoretical Quantiles
p values for Ljung-Box statistic
p value
0.8
0.4
0.0
5
10
15
20
25
30
35
LAG (H)

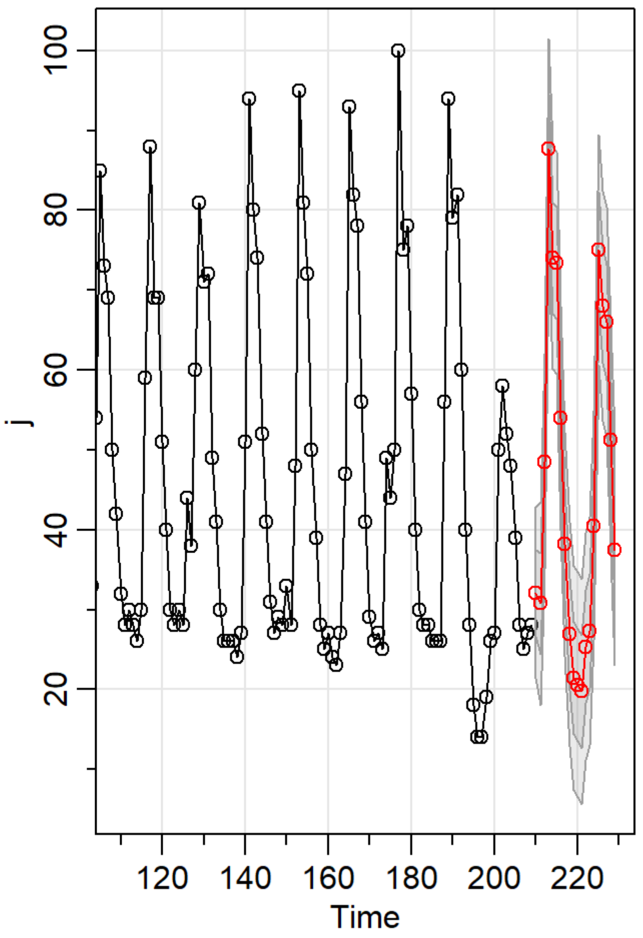
```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1     sar1     sma1
##       0.6122  -0.9804  -0.3004  -0.5669
## s.e.  0.0632   0.0248   0.1020   0.0885
##
## sigma^2 estimated as 28.88:  log likelihood = -614.21,  aic = 1238.43
##
## $degrees_of_freedom
## [1] 192
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1    0.6122 0.0632   9.6818  0.0000
## ma1   -0.9804 0.0248 -39.5930  0.0000
## sar1  -0.3004 0.1020  -2.9458  0.0036
## sma1  -0.5669 0.0885  -6.4067  0.0000
##
## $AIC
## [1] 5.982751
##
## $AICc
## [1] 5.983708
##
## $BIC
## [1] 6.061933
```

Forecast

同時繪製由方法一及方法二模型產生出的20步預測值。

```
#forecast
par(mfrow=c(1,2))
{ts.plot(j)
  sarima.for(j, 20, 1,1,1, 1,1,1, 12)}
```





```
## $pred
## Time Series:
## Start = 210
## End = 229
## Frequency = 1
##  [1] 32.10219 30.76912 48.52120 87.68252 74.00641 73.41295 53.98344 38.17639
##  [9] 26.93349 21.42606 20.56948 19.72938 25.35069 27.28604 40.47328 75.06566
## [17] 68.07741 65.96700 51.23890 37.51723
##
## $se
## Time Series:
## Start = 210
## End = 229
## Frequency = 1
##  [1] 5.373860 6.356527 6.721206 6.874215 6.944895 6.981128 7.001991 7.015544
##  [9] 7.025401 7.033284 7.040066 7.046214 7.116666 7.154087 7.176487 7.191601
## [17] 7.202963 7.212291 7.220470 7.227982
```

Efficiency

將資料切分成訓練集及驗證集，並計算以兩擬合後模型，套用至驗證集上的rmse。可得兩方法結果相近。

```
# loading packages
# install.packages('Metrics')
library(forecast)
library(Metrics)

# splitting data into train and valid sets
trainj = j[1:168]
validj = j[168:length(j)]

# training model
modelj = arima(trainj, order=c(1,0,0), season = list(order=c(0,1,2), period=12), method = 'ML')
model2j = arima(trainj, order=c(1,1,1), season = list(order=c(1,1,1), period=12), method = 'ML')

# model summary
summary(modelj)
```

```
##
## Call:
## arima(x = trainj, order = c(1, 0, 0), seasonal = list(order = c(0, 1, 2), period = 12),
##     method = "ML")
##
## Coefficients:
##          ar1     sma1    sma2
##       0.4975  -0.8782  0.3223
## s.e.  0.0701   0.1064  0.1005
##
## sigma^2 estimated as 17.43:  log likelihood = -449.23,  aic = 906.46
##
## Training set error measures:
##                     ME    RMSE      MAE        MPE    MAPE     MASE       ACF1
## Training set 0.3121991 4.02274 2.630221 -0.8370348 5.96596 0.272148 0.02505819
```

```
summary(model2j)
```

```
##
## Call:
## arima(x = trainj, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##          ar1     ma1     sar1     sma1
##       0.5093  -1.000  -0.3797  -0.4070
## s.e.  0.0706   0.028   0.1029   0.1107
##
## sigma^2 estimated as 17.83:  log likelihood = -449.67,  aic = 909.34
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.3003787 4.055592 2.741901 -2.306182 6.477847 0.2837035
##                    ACF1
## Training set 0.01383032
```

```
# forecasting
forecastj = predict(modelj,42)
forecastj$pred
```

```
## Time Series:
## Start = 169
## End = 210
## Frequency = 1
##  [1] 43.02180 31.15861 27.02933 27.98448 26.78220 30.19157 28.72107 50.55480
##  [9] 92.96048 79.56613 73.67179 51.39137 40.21723 29.35778 25.85163 27.38270
## [17] 25.70708 27.38483 27.44797 48.06969 94.05451 81.50913 75.37044 53.04043
## [25] 41.03761 29.76590 26.05466 27.48370 25.75733 27.40983 27.46041 48.07588
## [33] 94.05759 81.51066 75.37120 53.04081 41.03780 29.76599 26.05471 27.48373
## [41] 25.75734 27.40984
```

```
forecast2j = predict(model2j,42)
forecast2j$pred
```

```
## Time Series:
## Start = 169
## End = 210
## Frequency = 1
##  [1] 43.51928 31.99029 27.77684 29.10589 27.59324 30.51886 29.21451 50.41844
##  [9] 93.45282 80.29652 74.09020 52.21055 41.21346 30.35460 26.84086 28.54636
## [17] 26.53103 27.99784 28.72340 49.47844 93.64291 81.30738 75.93979 54.01500
## [25] 42.45492 31.34178 27.56247 29.12508 27.30062 29.32128 29.27618 50.20165
## [33] 93.93707 81.28993 75.60391 53.69626 42.34992 31.33332 27.65483 29.27170
## [41] 27.37477 29.18516
```

```
# evaluation
rmse(validj, forecastj$pred)
```

```
## [1] 18.55826
```

```
rmse(validj, forecast2j$pred)
```

```
## [1] 18.5386
```