

112學年度第一學期

計量經濟學實證報告

失業率與犯罪率之相關性

經碩一 112258017 徐于婷

## 一、研究動機

一直以來，台灣及國外都有很多研究失業率與犯罪率之間的關係，而失業率變高也反映著當下的經濟情況可能不是那麼穩定，因此可能也使那段時間社會較動盪，各類型犯罪案件可能隨之增加。尤其疫情之後造成的經濟衝擊，讓更多人重新探討經濟衝擊對犯罪率的影響，也讓我讀到一些國內外探討犯罪率與失業或者其他經濟變數相關性的實證報告，而我剛好也對這個議題蠻好奇的，藉由此次計量的實證報告來嘗試做這方面的議題。根據我讀到的文獻，這方面的實證研究雖然已經很多人做過了，但是所研究的樣本年代都已經較久遠，且文獻顯示出的結果並不是都一致，失業率與全般刑案犯罪率的關係，結果為正相關及無顯著相關的大概各佔一半，並且根據林明仁(2005)對於實證文獻的整理，有些美國實證也有出現負相關但不顯著的結果，因此在這次實證報告中，以台灣近十年較新的資料來做實證，來探討結果是否會不同。

## 二、文獻回顧

根據過去研究犯罪率的文獻當中，林明仁、劉仲偉(2005)以台灣1978-2003年縣市資料為例，以OLS最小平方法並加入三個工具變數的方式以及2SLS兩階段最小平方法，研究失業率及全般刑案發生率的相關性，變數選用則包含社會經濟變數：失業率、平均每戶收入、社會福利支出及教育程度等；嚇阻變數為刑案破獲率、警政支出等。實證結果顯示，在OLS最小平方法下，失業率對全般刑案發生率有顯著的正相關；嚇阻變數部分呈現顯著正相關的是警政支出，負相關的是破獲率。而兩階段最小平方法下，失業率同樣對全般刑案發生率有顯著正相關，破獲率及警政支出則與OLS的結果相同。

葉玫麟(2011)以1994-2007年的23個縣市資料來進行，研究方法為利用非線性的縱橫移轉平滑模型來進行，並且也以全般刑案發生率及其他分類案件的發生率作為解釋變數；解釋變數的部分則包含失業率、教育程度、可支配所得、刑案破獲率等。並納入前兩期的失業率作為模型所需的門檻變數。此實證研究的分析結果為：在失業率低於門檻值時，失業率的變動對犯罪率的影響較大；失業率已高於門檻值的情況下，對全般刑案犯罪率的影響則相對較小。文獻也以縱橫最小平方法來作分析比較，得到的結果與利用PSTR模型結果相同，因此在這兩篇文獻中，皆呈現出失業率與犯罪發生的正相關性。

## 三、資料來源與敘述統計

表一、2010/1-2023/8台灣地區月資料

變數名稱	定義	來源
全般刑案發生率 (cri)	每十萬人口中的全般刑案發生件數	警政統計查詢網
失業率(unem)	失業者佔勞動力比率	主計處
刑案破獲率(dec)	當期刑事案件中每百發生件數之破獲件數	警政統計查詢網

#### 一、被解釋變數

全般刑案發生率(cri)，將此作為被解釋變數最主要是因為全般刑案的發生與失業率的關係是目前實證中較不一致的部分，以分類的案件來看，像是竊盜、公共危險罪等等，在實證研究上都有較一致的結果，因此，此實證報告選擇以近十年的全般刑案發生率來作為此報告的被解釋變數，並且取對數進行分析。

#### 二、解釋變數

根據過去文獻顯示，在探討這類型議題時，主要影響犯罪率的兩大變數類別為：社會經濟變數以及嚇阻變數，嚇阻變數是Becker(1968)提出，人們會衡量成本及效用來決定是否執行犯罪行為，如犯罪後面臨的刑期、罰款或是政府為避免犯罪行為而實行的政策等。因此考量到我的資料筆數及必須納入的嚇阻變數後，選擇過去文獻中對犯罪率有顯著相關性的刑案破獲率來做為解釋變數。

而失業率為此次報告想探討的部分，社會經濟變數當中，過去許多文獻利用各種不同的經濟變數探討犯罪率，但此次報告主要想研究是近十年的失業率對犯罪率的關係是否和過去的結果有所不同，同樣因為資料筆數的關係，避免模型估計不準確，社會變數只納入主要想關心的變數失業率。

表二、敘述統計, 2010/1-2023/8

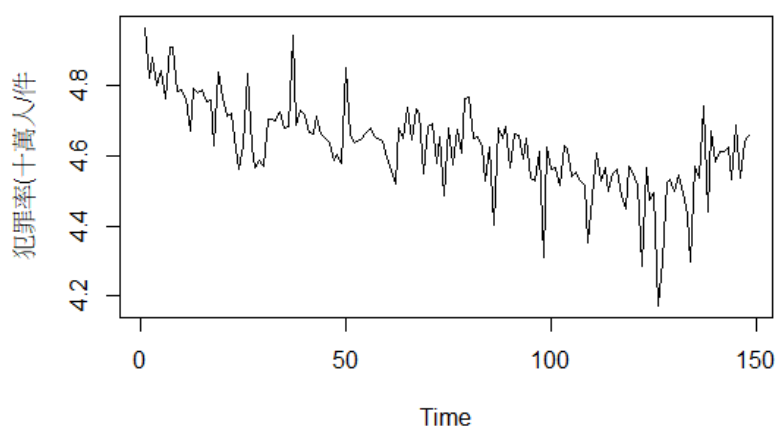
	樣本數	平均值	標準差	最小值	最大值
全般刑案發生率 (log)	148	4.627676	0.1272555	4.172694	4.965429
刑案破獲率	148	91.786757	7.2405536	70.79	108.83
失業率	148	3.913243	0.2709927	3.46	4.8

#### 四、研究方法

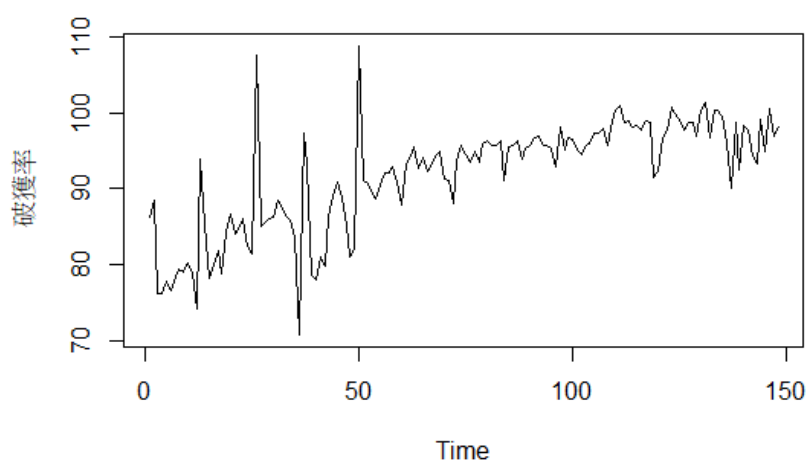
本實證報告採2010年1月到2023年8月的時間序列月資料來分析，研究方法利用OLS最小平方法，原始模型設定為式(1)。而因為時間序列資料大多都為非定態，因此進行迴歸分析之前，需先確認平穩性，並檢定自相關問題，才能進行。

$$\log(cri) = \beta_0 + \beta_1 dec + \beta_2 unem + e \quad (1)$$

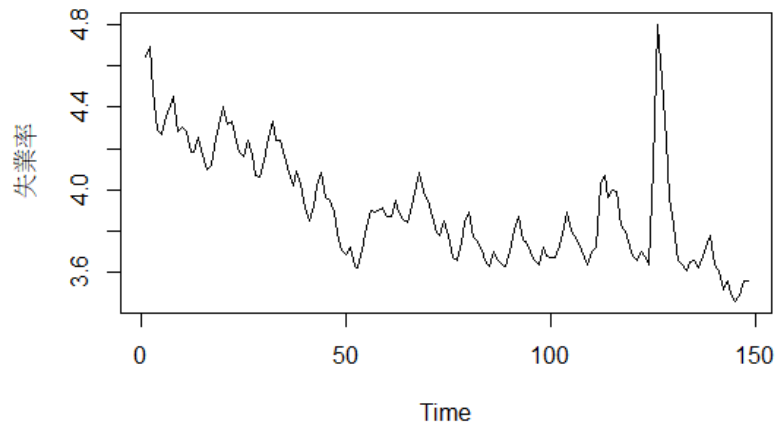
### 第一節、平穩性(單根檢定)



圖一、取對數後的犯罪率時間趨勢圖



圖二、破獲率時間趨勢圖



圖三、失業率時間趨勢圖

從三個變數的時間趨勢圖可以初步看出資料為非定態，因此進一步利用ADF (Augmented Dickey-Fuller) 來確認資料的確具有單根。Dickey-Fuller 單根檢定模型為：

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \dots + \delta_k \Delta y_{t-k+1} + \varepsilon_t \quad (2)$$

作法先作差分及整理出各變數的資料矩陣，設定滯後期數後，先分別計算三個變數估計係數以及標準誤，再將係數除以標準誤，計算出ADF統計量，其結果如表三：

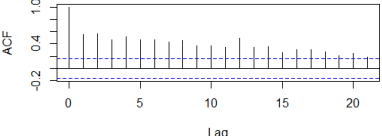
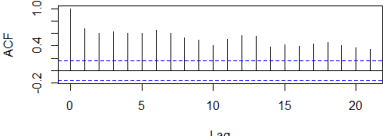
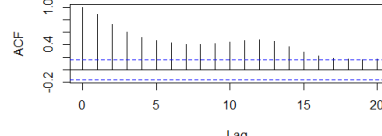
變數	估計係數 $\gamma$	標準誤	ADF統計
犯罪率落後一期(cri_lag)	-0.4639972	0.15176509	-3.057338
破獲率落後一期(dec_lag)	-0.3958445	0.15173398	-2.608806
失業率落後一期(unem_lag)	-0.1764643	0.05457112	-3.233657

表三、ADF統計結果

呈現的結果及樣本數來根據Dickey-Fuller t分配的查表，三個變數的序列資料皆接受虛無假設，也就是為非定態，因此需要將資料進一步做差分。

## 第二節、自相關檢定

接下來因時間序列資料通常與自己的落後期具有高度相關性，所以在進行迴歸分析前，需要透過計算資料與期落後期的自相關係數。時間越久自相關係數將會越小。檢定的方法利用ACF(Autocorrelation Function)來找出資料與幾期落後期具高度相關。作法為先計算資料減掉平均後的值，依設定的落後期數將資料分成兩區段，計算皮爾森相關係數。計算結果與直接執行ACF檢定相同，如下表：

犯罪率(log(cri))	破獲率(dec)	失業率(unem)
<div>by_hand      acf</div> <div>lags.0   1.0000000 1.0000000</div> <div>lags.1   0.5573629 0.5573629</div> <div>lags.2   0.5715143 0.5715143</div> <div>lags.3   0.4719794 0.4719794</div> <div>lags.4   0.5227410 0.5227410</div> <div>lags.5   0.4654874 0.4654874</div> <div>lags.6   0.4634674 0.4634674</div> <div>lags.7   0.4293379 0.4293379</div> <div>lags.8   0.4540750 0.4540750</div> <div>lags.9   0.3667324 0.3667324</div> <div>lags.10 0.3646102 0.3646102</div>	<div>by_hand      acf</div> <div>lags.0   1.0000000 1.0000000</div> <div>lags.1   0.6756704 0.6756704</div> <div>lags.2   0.6085103 0.6085103</div> <div>lags.3   0.6269722 0.6269722</div> <div>lags.4   0.6091804 0.6091804</div> <div>lags.5   0.6096539 0.6096539</div> <div>lags.6   0.6531469 0.6531469</div> <div>lags.7   0.5987647 0.5987647</div> <div>lags.8   0.5333195 0.5333195</div> <div>lags.9   0.4906555 0.4906555</div> <div>lags.10 0.4111338 0.4111338</div>	<div>by_hand      acf</div> <div>lags.0   1.0000000 1.0000000</div> <div>lags.1   0.8878323 0.8878323</div> <div>lags.2   0.7235084 0.7235084</div> <div>lags.3   0.6004520 0.6004520</div> <div>lags.4   0.5225994 0.5225994</div> <div>lags.5   0.4649284 0.4649284</div> <div>lags.6   0.4333479 0.4333479</div> <div>lags.7   0.4091101 0.4091101</div> <div>lags.8   0.4029840 0.4029840</div> <div>lags.9   0.4155464 0.4155464</div> <div>lags.10 0.4403103 0.4403103</div>
		

表四、ACF test皆有至少15期的自相關

## 第三節、差分

經過對資料進行平穩檢定及自相關檢定後，證明資料非定態及自相關，因此對三個變數：取對數後的犯罪率、破獲率及失業率進行一階差分。迴歸模型表示如下：

$$dec_{it} = \beta_0 + \beta_1 dec_{it-1} + \beta_2 deunem_{it} + e_{it} \quad (3)$$

## 第四節、差分後平穩性及自相關

差分後為確保資料已平穩，再做一次ADF檢定，如下表五，從P值判斷，為小於0.05，可確定三個序列資料已是定態。表六中，雖然犯罪率與落後一期還有到-0.5的自相關係數，但綜合以上結果，自相關性大幅降低，不再進行差分。

變數	ADF統計量	P 值
犯罪率落後一期(cri_lag)	-7.4704	0.01
破獲率落後一期(dec_lag)	-9.6958	0.01
失業率落後一期(unem_lag)	-6.3788	0.01

表五、差分後ADF檢定結果

	犯罪率(decric)	破獲率(dedec)	失業率(deunem)
落後一期	-0.52	0.39	0.29
落後二期	0.13	-0.15	-0.18
落後三期	-0.17	-0.06	-0.2

表六、差分後ACF檢定

## 第五節、異質性檢定

資料去除掉時間趨勢後，異質變異的問題可能還是存在在模型中，接下來將利用 Breusch-Pagan Test來檢定異質性是否存在( $H_0$ :同質變異)。檢定步驟為以式(3)計算出殘差，並取殘差平方，接下來以殘差平方為被解釋變數，差分後的破獲率及失業率作為解釋變數，進而估計係數值以及計算R squared。最後再算出BP test統計量及P值。

BP 統計量	P 值
40.92079	1.300656e-09***
註:*** p < 0.01, ** p < 0.05, *p < 0.1, 具有異質性	

表七、BP檢定結果

## 五、實證結果

在資料差分為定態後，資料進行BP檢定仍具有異質性，因此最後做OLS估計前需要利用White Estimator調整異質性問題，讓迴歸模型估計能更加準確。表八是利用White Estimator後迴歸模型參數變異數估計出來的結果：

變數	By hand	By package
Intercept	8.484595e-05	8.484595e-05
破獲率(dedec)	2.211225e-06	2.211225e-06
失業率(deunem)	1.068632e-02	1.068632e-02

表八、White 估計結果

經過將資料轉為定態及異質性的調整後，利用OLS估計出來的結果如下表九，實證結果顯示，只有破獲率對全般刑案犯罪率有顯著正相關，係數為:0.0060902，而報告主要想探討的失業率則跟全般刑案犯罪率，從P值來看，兩者沒有顯著相關性，此結果與過去一些文獻的實證相同。

最後以Wald Test來檢驗迴歸模型中，各個解釋變數的係數是否有意義，可以從中知道，所選擇的解釋變數估計出的係數是否對被解釋變數具有顯著性，從下表十可以看出，Wald Test檢定出來的P值小於0.05，可以拒絕虛無假設，也就是係數非為0，迴歸模型中的係數對被解釋變數來說都是重要的，模型具有一定的解釋力。

變數	OLS估計結果	std.error	t value	p value
Intercept	-0.0037777	0.0092112	-0.4101238	0.6823123
破獲率(dedec)	0.0060902	0.0014870	4.0955657	0.0000693***
失業率(deunem)	-0.1650549	0.1033746	-1.5966672	0.1124874

註:\*\*\* p < 0.01, \*\* p < 0.05, \*p < 0.1

表九、迴歸結果



Wald test統計	8.937682
P value	0.0002186068***

表十、Wald test結果

## 六、建議與未來碩論方向

從這次利用時間序列資料作實證分析，了解到時間序列資料的處理過程繁複，利用OLS分析前，需要從自相關到平穩檢定，有時還會需要處理異質性。除此之外，根據文獻，其實有些嚇阻變數可能會有內生性問題，因為犯罪率較高的地方，政府可能在加強警政方面上會有更多措施，例如，警政支出；甚至是報告中放入的破獲率都有可能因為地區本身犯罪率高而破獲率也高，因而產生正相關。所以未來再次做這方面的議題時，需要更嚴謹，要改進的地方首先為資料筆數要再增加，然後增加解釋變數個數及因應內生性問題來調整嚇阻變數。模型方面也可以嘗試像是VAR向量自我迴歸模型，就可以考量到內生性來處理資料；也可以深入研究Granger因果關係檢定的方法，若以因果關係檢定來分析，就可以看出「統計上」是否會影響被解釋變數，也許能夠建構出更完整的模型，結果更加準確。

經過這次的實證報告，在未來碩論想要嘗試不同型態的資料來做實證，目前對碩論的想法是做個體實證，對其中的勞動經濟也蠻有興趣，感覺能做的議題會更生活化，所以未來可能會面對的是資料應該是橫斷面資料居多，這次學到處理異質性問題，希望能加強處理橫斷面資料的方法。

### 參考文獻：

林明仁、劉仲偉(2006),「失業真的會導致犯罪嗎?以台灣1978年至2003年縣市資料為例」經濟論文叢刊, 34(4), 445-482。

施佩珊(2012),「婦女勞動參與與青少年犯罪-台灣實證研究」

葉玫麟(2011),「台灣地區犯罪率與失業率之實證研究-非線性模型的應用」

方柏翔(2009),「台灣暴力犯罪與失業率之關聯性分析-向量自迴歸模型之運用」

## 附錄:指令

### ● 讀取資料加敘述統計

```
T <- read.csv("month.csv", header=TRUE, check.names=FALSE) #2010
options(max.print = 10000)
T
y <- ("cri"= T[, 2])
y <- as.matrix(y)
y
x <- T[, 3:4]
x <- as.matrix(x)

#敘述統計
##平均
mean_y <- sum("cri"=log(y))/length(log(y))
mean_x2 <- sum("dec"=x[, 2])/148
mean_x3 <- sum("unem"=x[, 3])/148
##標準差
sd(log(y))
sd(x[, 2])
sd(x[, 3])
summ <- cbind("nobs"= c(length(y), length(x[, 2]), length(x[, 3])),
              "mean"= c(mean_y, mean_x2, mean_x3),
              "sd"= c(sd(log(y)), sd(x[, 2]), sd(x[, 3])),
              "min"= c(min(log(y)), min(x[, 2]), min(x[, 3])),
              "max"= c(max(log(y)), max(x[, 2]), max(x[, 3])))

#放入截距
x <- cbind(rep(1), x)
x
beta <- solve(t(x) %*% x) %*% t(x) %*% y
beta
lm_f <- lm(y~x)
summary(lm_f)
```

### ● 平穩檢定(以差分前作代表)

```
#ADF test
k <- 6 ##設定6:(會有五個落後期做為差分後的解釋變數)
##失業率
adf.test(x3)
x3 <- x[, 3]
x3
adf <- function(x3, k=NULL) { #處理進行ADF的序列資料和滯後期數K
  if(is.null(k)) {
    k <- trunc((length(x3) - 1)^(1/3))
    k <- k + 1
  }
  if(k < 0) {
    k <- trunc((length(x3) - 1)^(1/3))
    k <- k + 1
  }
  adf_x3 <- diff(x3)
  n <- length(adf_x3)
  z <- embed(adf_x3, k) #轉換序列資料 by設定的lag order
  diff_unemt <- z[, 1] ##被解釋變數(差分後)
  unem_lag <- x3[k:n] #解釋變數:被解釋變數差分前k:n期的資料
  tt <- k:n #時間
  if(k > 1){
    d_unem_lag <- z[, 2:k] #解釋變數:差分過後的落後期資料(5個落後期)
    adf_x <- cbind(unem_lag, "inter"=1, tt, d_unem_lag)
    adf_be <- solve(t(adf_x) %*% adf_x) %*% t(adf_x) %*% diff_unemt
  } else {adf_x <- cbind(unem_lag, "inter"=1, tt)
    adf_be <- solve(t(adf_x) %*% adf_x) %*% t(adf_x) %*% diff_unemt
  }
}

#殘差
adf_eh <- diff_unemt - adf_x %*% adf_be #
residual_variance <- sum(adf_eh^2) / (length(tt) - ncol(adf_x)) #殘差變異數
XTX_inverse <- solve(t(adf_x) %*% adf_x)
#unem_lag標準誤
element_11 <- XTX_inverse[1, 1]
unemlag_se <- sqrt(residual_variance * element_11)
cat("unem落後期係數的標準誤差:", unemlag_se, "\n")
STAT <- adf_be[1, 1]/unemlag_se
TBL <- cbind(
  "1%" = c(-4.38, -4.15, -4.04, -3.99, -3.98, -3.96),
  "5%" = c(-3.60, -3.50, -3.45, -3.43, -3.42, -3.41))
rownames(TBL) <- paste0("T = ", c(25, 50, 100, 250, 500, Inf))
cat(strrep("#", 22))
cat("\ncritical value table\n")
print(TBL)
cat(strrep("#", 22))
cat("\n")
cat("T = ", n, "\n\n")
#which:返回向量
num <- which(n < c(25, 50, 100, 250, 500, Inf))[1] #看樣本數滿足哪個條件
num2 <- which(STAT < TBL[num, ]) #看是否小於相應表格中的值
if(length(num2) == 0) {
  cat("cannot reject H0: time series unem is non-stationary\n")
  cat("STAT: ", STAT, "\n")
} else {
  num3 <- num2[1]
  cat("REJECT H0: time series is stationary\n")
  cat("with significant level", colnames(TBL)[num3], "\n")
  cat("STAT: ", STAT, "\n")
}
```

```
##破獲率
x2 <- x[, 2]
adf <- function(x2, k=NULL) { #處理進行ADF的序列資料和滯後期數K

  if(is.null(k)) {
    k <- trunc((length(x2) - 1)^(1/3))
    k <- k + 1
  }

  if(k < 0) {
    k <- trunc((length(x2) - 1)^(1/3))
    k <- k + 1
  }
}
adf_x2 <- diff(x2)
n <- length(adf_x2)
z2 <- embed(adf_x2, k) #轉換序列資料w/落後期數
diff_dec <- z2[, 1] ##被解釋變數
dec_lag <- x2[k:n] #被解釋變數差分前k:n 的資料
tt <- k:n #時間
if(k > 1){
  d_dec_lag <- z2[, 2:k] ##差分後落後期
  adf_x2 <- cbind( dec_lag, "inter"=1, tt, d_dec_lag )
  adf_be2 <- solve(t(adf_x2) %*% adf_x2) %*% t(adf_x2) %*% diff_dec
} else {adf_x2 <- cbind(dec_lag, "inter"=1, tt)
adf_be2 <- solve(t(adf_x2) %*% adf_x2) %*% t(adf_x2) %*% diff_dec
}
#殘差
adf_ehdec <- diff_dec - adf_x2 %*% adf_be2 #same
edec_variance <- sum(adf_ehdec^2) / (length(tt) - ncol(adf_x2))
XTX_inverse <- solve(t(adf_x2) %*% adf_x2)
```

```
#xt1標準誤
element_11 <- XTX_inverse[1, 1]
dec_lag_se <- sqrt(edec_variance * element_11)
dec_lag_se
# 印出結果
cat("第", 1, "個係數的標準誤差:", dec_lag_se, "\n")
STAT3 <- adf_be2[1, 1]/dec_lag_se
TBL <- cbind(
  "1%" = c(-4.38, -4.15, -4.04, -3.99, -3.98, -3.96),
  "5%" = c(-3.60, -3.50, -3.45, -3.43, -3.42, -3.41))
rownames(TBL) <- paste0("T = ", c(25, 50, 100, 250, 500, Inf))
cat(strrep("#", 22))
cat("\ncritical value table\n")
print(TBL)
cat(strrep("#", 22))
cat("\n")
cat("T = ", n, "\n\n")
num <- which(n < c(25, 50, 100, 250, 500, Inf))[1]
num2 <- which(STAT3 < TBL[num, ])
if(length(num2) == 0) {
  cat("cannot reject H0: time series dec is non-stationary\n")
  cat("STAT3: ", STAT3, "\n")
} else {
  num3 <- num2[1]
  cat("REJECT H0: time series is stationary\n")
  cat("with significant level", colnames(TBL)[num3], "\n")
  cat("STAT3: ", STAT3, "\n")
}
```

```
####犯罪率
ly <- log(y)
adf <- function(ly, k=NULL) { #處理進行ADF的序列資料和滯後期數K

  if(is.null(k)) {
    k <- trunc((length(ly) - 1)^(1/3))
    k <- k + 1
  }

  if(k < 0) {
    k <- trunc((length(ly) - 1)^(1/3))
    k <- k + 1
  }
}
adf_ly <- diff(ly)
n <- length(adf_ly)
z1 <- embed(adf_ly, k) #轉換序列資料w/落後期數
diff_cri <- z1[, 1] #ADF的被解釋變數:差分後
cri_lag <- ly[k:n] #差分前k期到n期的資料
tt <- k:n #時間
if(k > 1){
  d_cri_lag <- z1[, 2:k]
  adf_x1 <- cbind( cri_lag, "inter"=1, tt, d_cri_lag )
  adf_be1 <- solve(t(adf_x1) %*% adf_x1) %*% t(adf_x1) %*% diff_cri
} else {adf_x1 <- cbind(cri_lag, "inter"=1, tt)
adf_be1 <- solve(t(adf_x1) %*% adf_x1) %*% t(adf_x1) %*% diff_cri
}##係數結果相同
#殘差
adf_ehcri <- diff_cri - adf_x1 %*% adf_be1 #same #殘差
ecri_variance <- sum(adf_ehcri^2) / (length(tt) - ncol(adf_x1))
XTX_inverse <- solve(t(adf_x1) %*% adf_x1)
#cri標準誤
```

```
element_11 <- XTX_inverse[1, 1]
cri_lag_se <- sqrt(ecri_variance * element_11)
# 印出結果
cat("cri落後期係數的標準誤差:", cri_lag_se, "\n")
STAT2 <- adf_be1[1, 1]/cri_lag_se
STAT2
TBL <- cbind(
  "1%" = c(-4.38, -4.15, -4.04, -3.99, -3.98, -3.96),
  "5%" = c(-3.60, -3.50, -3.45, -3.43, -3.42, -3.41))
rownames(TBL) <- paste0("T = ", c(25, 50, 100, 250, 500, Inf))
cat(strrep("#", 22))
cat("\ncritical value table\n")
print(TBL)
cat(strrep("#", 22))
cat("\n")
cat("T = ", n, "\n\n")
num <- which(n < c(25, 50, 100, 250, 500, Inf))[1]
num2 <- which(STAT2 < TBL[num, ])
if(length(num2) == 0) {
  cat("cannot reject H0: time series: cri is non-stationary\n")
  cat("STAT2: ", STAT2, "\n")
} else {
  num3 <- num2[1]
  cat("REJECT H0: time series is stationary\n")
  cat("with significant level", colnames(TBL)[num3], "\n")
  cat("STAT2: ", STAT2, "\n")
}
```

```
> adf_data <- data.frame(
+   coef=c(adf_be1[1,1], adf_be2[1,1], adf_be[1,1]),
+   標準誤=c(cri_lag_se, dec_lag_se, unemlag_se),
+   ADF統計=c(adf_be1[1,1]/cri_lag_se, adf_be2[1,1]/dec_lag_se, adf_be[1,1]/unemlag_s
+ ),
+   ADF套件=c(adf.test(ly)$statistic, adf.test(x2)$statistic, adf.test(x3)$statisti
+ c))
> adf_data
      coef      標準誤  ADF統計  ADF套件
cri_lag -0.4639972 0.15176509 -3.057338 -3.057338
dec_lag -0.3958445 0.15173398 -2.608806 -2.608806
unem_lag -0.1764643 0.05457112 -3.233657 -3.233657
```

- ACF檢定(以差分前作代表)

```
#犯罪率#犯罪率以取對數來分析[
res <- acf(log(y))
n <- length(y)
y0 <- log(y)-mean_y
nlags <- 21
res2 <- sapply(1:nlags, \i) { #對每個滯後期數迭代
  a <- y0[seq_len(n-i)] #1~(n-i)的資料(xi-mean)的值
  b <- y0[-seq_len(i)] #剩下的值
  sum(a * b) / sum(y0 * y0) #算自相關係數
})
RES <- cbind("by_hand" = c(1, res2),
            "acf" = res$acf[,1][1:(nlags+1)])

rownames(RES) <- paste0("lags.", 0:nlags)
RES
```

```
#失業率
res_unem2 <- acf(x[, 3])
n <- length(y)
x3 <- x[, 3]-mean_x3
nlags <- 17
res_unem <- sapply(1:nlags, \i) {
  a3 <- x3[seq_len(n-i)]
  b3 <- x3[-seq_len(i)]
  sum(a3 * b3) / sum(x3 * x3)
})
RES <- cbind("by_hand" = c(1, res_unem),
            "acf" = res_unem2$acf[,1][1:(nlags+1)])

rownames(RES) <- paste0("lags.", 0:nlags)
RES
```

```
#破獲率
res_dec2 <- acf(x[, 2])
n <- length(y)
x2 <- x[, 2]-mean_x2
nlags <- 21
res_dec <- sapply(1:nlags, \i) {
  a2 <- x2[seq_len(n-i)]
  b2 <- x2[-seq_len(i)]
  sum(a2 * b2) / sum(x2 * x2)
})
RES <- cbind("by_hand" = c(1, res_dec),
            "acf" = res_dec2$acf[,1][1:(nlags+1)])

rownames(RES) <- paste0("lags.", 0:nlags)
RES
```

- 差分

```
##差分
de_y <- diff(log(y)) #犯罪率差分
time_axis_diff <- seq_along(de_y)
plot(de_y, type="l", xlab="Time", ylab="crime")
de_x <- diff(x[, 2:3]) #失業率和破獲率差分
de_x <- cbind("Inte" = 1, de_x)
beh <- solve(t(de_x) %*% de_x) %*% t(de_x) %*% de_y
lm_d <- lm(de_y ~ 0 + de_x)
summary(lm_d)
```

- 異質性檢定

```
eh_d <- residuals(lm_d)
#手刻殘差
ehat <- de_y - de_x %*% beh
eh2 <- ehat^2
be3 <- solve(t(de_x) %*% de_x) %*% t(de_x) %*% eh2
be3
#做R squared
yhat <- de_x %*% be3
R2 <- drop(crossprod(yhat)/crossprod(eh2))
R2
bp1 <- nrow(de_x)*R2
bp1
significance_level <- 0.05
p_value <- 1 - pchisq(bp1, df = 2)
p_value #有異質性
```

```
#套件對照
aux_model2 <- lm(eh_d^2 ~ 0 + de_x)
summary(aux_model2)
bp_statistic2 <- nobs(lm_d) * summary(aux_model2)$r.squared
bp_statistic2
p_value2 <- 1 - pchisq(bp_statistic2, df = 2)
p_value2
```

```
> my_data <- data.frame(
+   Item = c("By hand", "By package"),
+   BPstat = c(bp1, bp_statistic2),
+   Pvalue = c(p_value, p_value2))
> my_data
      Item    BPstat      Pvalue
1 By hand 40.92079 1.300656e-09
2 By package 40.92079 1.300656e-09
```

- White估計式-迴歸-Wald test

```
#under異質性#用white estimator
eh2 <- ehat^2
box <- 0
i <- 1
for(i in 1:nrow(de_x)){
  box <- box+eh2[i]*de_x[i, ] %*% t(de_x[i, ]) #white估計中的sum(殘差平方*X*X')
}
XtX <- t(de_x) %*% de_x
HC <- solve(XtX) %*% box %*% solve(XtX)
v <- diag(solve(XtX) %*% box %*% solve(XtX)) #white估計後的beta變異數
print(v)
#套件對照
library("lmtest")
library(sandwich)
v2 <- diag(vcovHC(lm_d, type="HC0")) ##vcov: white估計用來計算共變異數矩陣
v2
cbind("by_hand"= v, "package"= v2)
coeftest(lm_d, vcov = vcovHC(lm_d, type="HC0"))
##

#
RES <- cbind("Estimate"=c(beh),
            "std.Error"=sqrt(v),
            "t value"=c(beh)/sqrt(v),
            "Pr(>|t|)"= 2-2*pt(abs(c(beh)/sqrt(v)), df=nrow(de_x)))
round(RES, 7) ##模型結論

#wald test
R <- diag(ncol(de_x))
R <- R[-1, ]
r <- cbind(rep(0, ncol(de_x)-1))
q <- nrow(R)
N <- nrow(de_x)
wald.test.HC <- t(R %*% beh -r) %*% solve(R %*% HC %*% t(R)) %*% (R %*% beh -r)/q
wald.test.HC
1-pf(wald.test.HC, q, N-q)
```