UK Macro History

Peter Nash

14 December 2020

UK Macro Economic Relationships

We look at a subset of the full series between 1885 and 1985 with the following variables

- U_t: percent unemployment rate. Col ;
- P_t : PGDP: GDP deáator, 2013=100.
- Q_t: real UK GDP at market prices, geographically consistent estimate based on post 1922 borders. £ mn Chained Volume measure, 2013 prices. Col A1.B.
- RS_t: short interest rates, percent per annum, (Bank Rate).
- RL_t: long interest rates, percent per annum, (Consol/10 year debt) Col

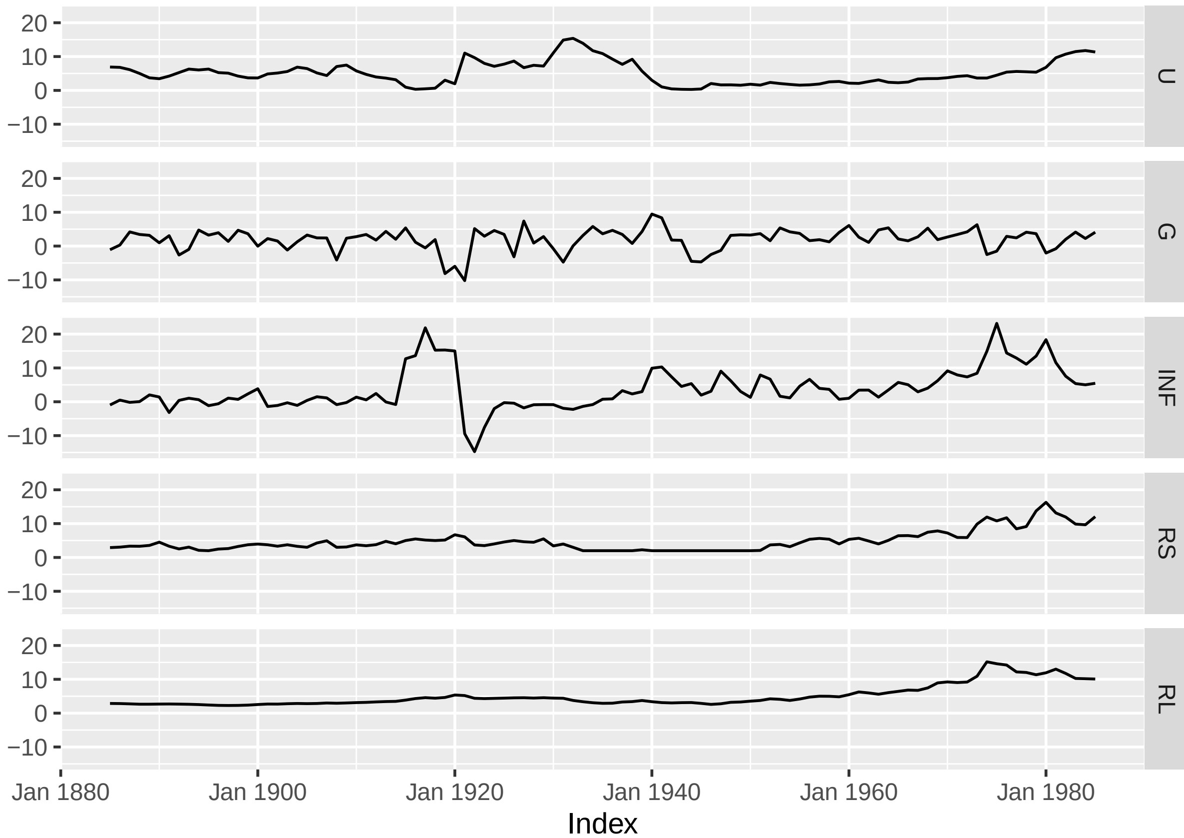
and their subsequent transformations

- log GDP: LQ_t
- log GDP deflator: LPt
- inflation: $INFt = 100($LP_t$ LP_{t-1})$;
- growth: $G_t = 100(LQ_t LQ_{t-1}$

Expected relationships

We would expect to see cycles of growth, decreasing unemployment, potentially encouraged by easier monetary policy and lower short term interest rates. In due course this may subsequently lead to inflationary pressures as slack in the labour supply decreases and wages increase. Monetary policy may induce increases in short term interest rates and subsequent cooling of economic growth. We would also expect inflationary pressures to lead to higher long term interest rates. We may see inflationary shocks or high periods of inflation leading to poor economic growth and higher unemployment.

```
autoplot.zoo(macro.subset)
```



Summary Statistics & Commentary

```
summary(macro.subset)
```

```
##      Index            U                    G                   INF
##  Min.   :1885   Min.   : 0.2835   Min.   :-10.2153   Min.   :-14.7434
##  1st Qu.:1910   1st Qu.: 2.3972   1st Qu.:  0.9712   1st Qu.: -0.1722
##  Median :1935   Median : 4.3887   Median :  2.6406   Median :  2.3143
##  Mean   :1935   Mean   : 5.0910   Mean   :  1.9688   Mean   :  3.7673
##  3rd Qu.:1960   3rd Qu.: 6.8727   3rd Qu.:  3.9406   3rd Qu.:  6.6286
##  Max.   :1985   Max.   :15.3873   Max.   :  9.4607   Max.   : 23.1675
##        RS                RL
##  Min.   : 2.000   Min.   : 2.264
##  1st Qu.: 3.000   1st Qu.: 2.904
##  Median : 3.959   Median : 3.756
##  Mean   : 4.824   Mean   : 5.061
##  3rd Qu.: 5.496   3rd Qu.: 5.458
##  Max.   :16.301   Max.   :15.173
```

Unemployment averages at 5% over the period with an average growth rate of 1.96%. The mean of inflation is 3.76% with the short and long term interest rates averaging at 4.8% and 5% respectively. We see peaks of inflation at 23% during the 1970s oil crisis, similar sustained periods of high inflation during WWI and unemployment at 15% following the great depression. Short term and long term interest rates show significant increases during the 1970s and onwards.

```r
cor(macro.subset)
```

```
##              U          G        INF         RS         RL
## U    1.0000000 -0.0883304 -0.4006868  0.1359800  0.1201737
## G   -0.0883304  1.0000000 -0.1001583 -0.0861742 -0.0204763
## INF -0.4006868 -0.1001583  1.0000000  0.5439262  0.5781197
## RS   0.1359800 -0.0861742  0.5439262  1.0000000  0.9114121
## RL   0.1201737 -0.0204763  0.5781197  0.9114121  1.0000000
```

Unrestricted Model

We consider the unrestricted model $U_t = \alpha_0 + \alpha_1 U_{t-1} + \alpha_2 U_{t-2} + \beta_0 LQ_t + \beta_1 LQ_{t-1} + \beta_2 LQ_{t-2} + \gamma t + \epsilon_{1t}$ and observe the following

```r
ur.ardl.model <- dyn$lm(U ~ stats::lag(U, k=1) +
                      stats::lag(U, k=2) +
                        LQ +
                      stats::lag(LQ, k=1) +
                      stats::lag(LQ, k=2) + trend, data = macro.series)

pander(summary(ur.ardl.model))
```

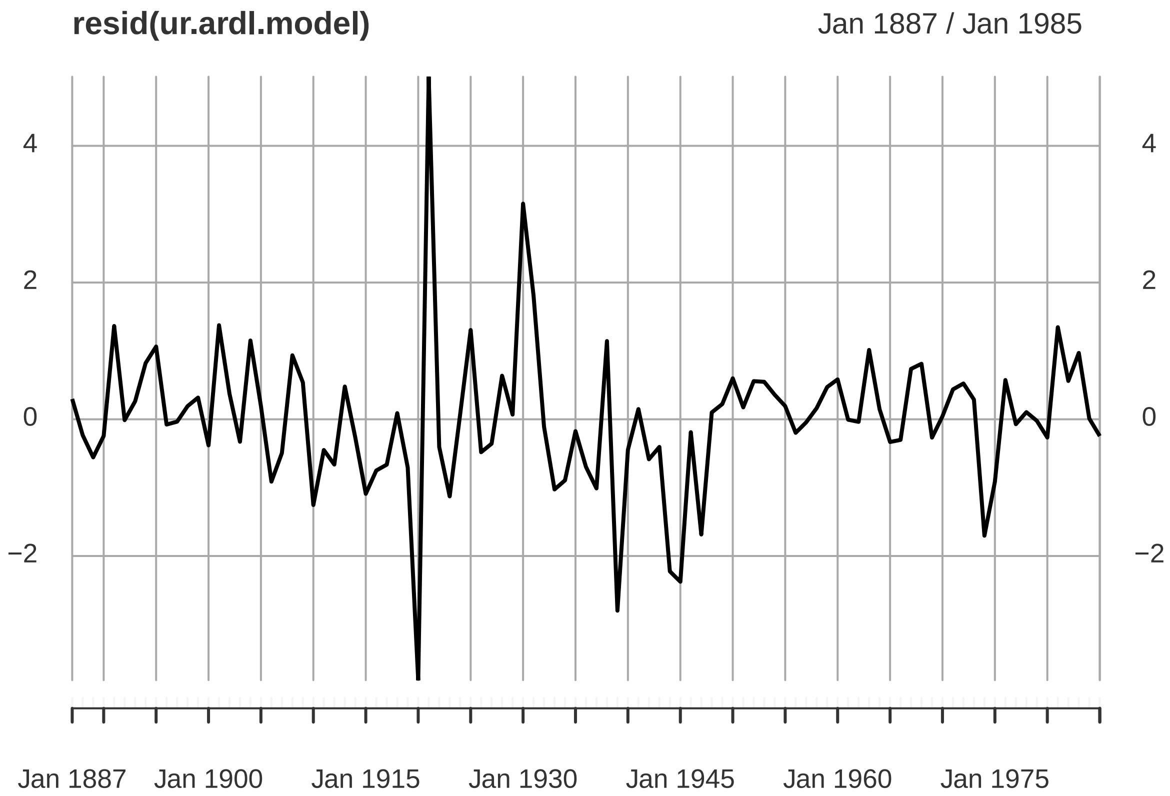
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-4.452	12.12	-0.3672	0.7143
stats::lag(U, k = 1)	0.9582	0.1052	9.112	1.689e-14
stats::lag(U, k = 2)	0.03401	0.1097	0.3099	0.7574
LQ	-30.79	3.852	-7.995	3.692e-12
stats::lag(LQ, k = 1)	26.67	7.327	3.639	0.000451
stats::lag(LQ, k = 2)	4.535	5.253	0.8632	0.3902
trend	0.001319	0.02005	0.06581	0.9477

Table 2: Fitting linear model: dyn(U ~ stats::lag(U, k = 1) + stats::lag(U, k = 2) + LQ + stats::lag(LQ, k = 1) + stats::lag(LQ, k = 2) + trend)

Observations	Residual Std. Error	R^2	Adjusted R^2
99	1.1	0.9049	0.8987

We note the coefficient of 1 period lagged unemployment, $\alpha_1 = 0.96$, as individually significant, suggesting a 1% increase in lagged unemployment will lead to a 0.96% increase in the following period. We also note that the log GDP and lagged log GDP LQ as significant, with a 1% increase in GDP proposing a -0.3% reduction in unemployment. However, we note that an increase in lagged GDP leading to a increase in unemployment, perhaps signalling some behaviour of the business cycle. The trend, t nor 2 period lagged unemplpoyment, U_{t-2} and 2 period lagged log GDP, LQ_{t-1} are deemed to be significant.

```r
plot(resid(ur.ardl.model))
```



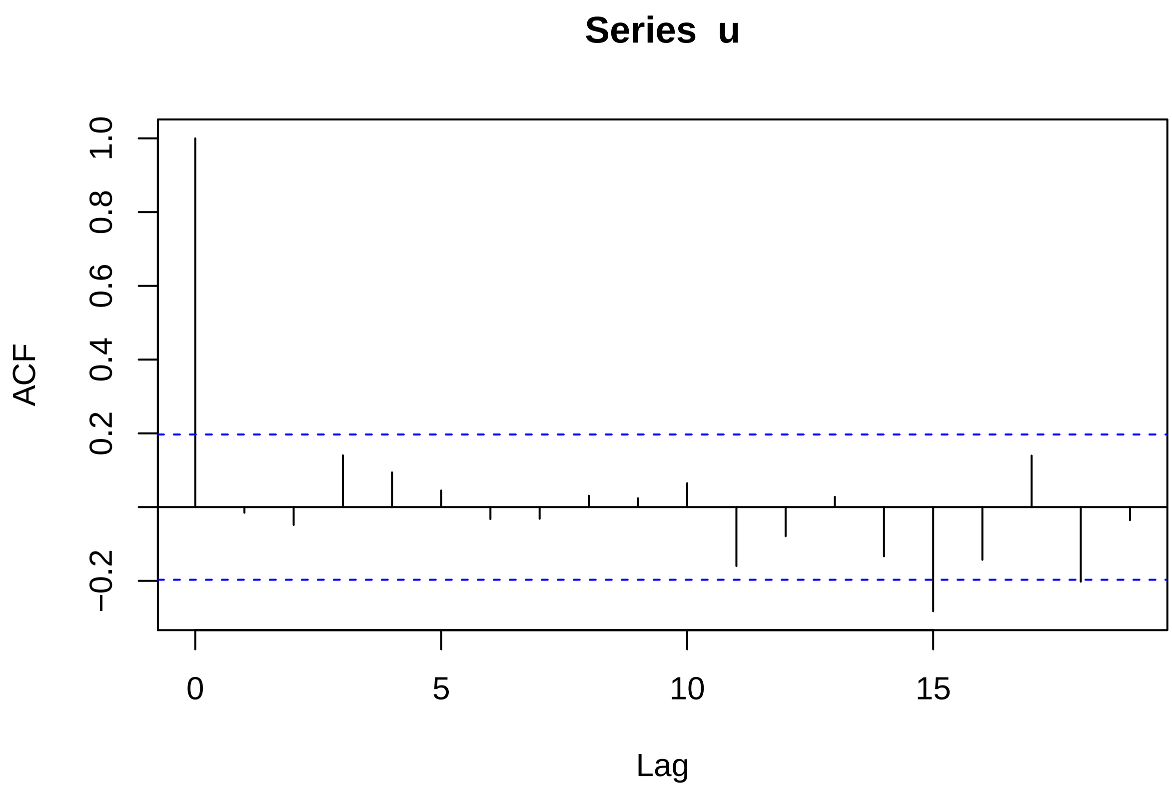
We can observe potential heteroskedasticity in the residuals with greater variance in the early to mid 20th century and some notable outliers.

Diagnostic tests

```
u <- resid(ur.ardl.model)
print(paste("DW:", durbinWatsonTest(as.vector(u))))
```

```
## [1] "DW: 2.02922619453604"
```

```
acf(u)
```



We note that DW is close to 2 (2.03) so no serial correlation. We can also observe this visually in the included ACF chart. The inclusion of two lagged terms appears to take care of any serial correlation concerns.

We test for heteroskedasticity using $\hat{u}_t^2 = \alpha + b' z_t + v_t$ using the hypothesis that $b' = 0$. In this case we use $z_t = x_t$

```
u2 <- u * u
summary(dyn$lm(u2 ~ stats::lag(U, k=1) +
                 stats::lag(U, k=2) +
                 LQ +
                 stats::lag(LQ, k=1) +
                 stats::lag(LQ, k=2) + trend, data = macro.series))$fstatistic
```

```
##    value    numdf    dendf
## 10.85578  6.00000 92.00000
```

When we retrieve the F-statistic we see that this is 6.29 which means we can reject the null hypothesis that $b' = 0$ and of homoskedasticity and constant variance. Given what we observe visually in the residuals with differing regimes throughough the sample, a further variance ratio test (Goldfeld-Quandt) might be warranted

Note, However, if we perform a reset test $\hat{u}_i = \alpha + b' \hat{y}_i^2 + \epsilon_i$ we observe different results and we fail to reject the null hypothesis of homoskedasticity

```
yhat <- ur.ardl.model$fitted.values

pandoc.table(summary(lm(as.vector(u) ~ (yhat * yhat))))
```

```
##
## ------------------------------------------------------------
##             call                          terms
## ----------------------------- ------------------------------
##  lm(formula = as.vector(u) ~   as.vector(u) ~ (yhat * yhat)
##       (yhat * yhat))
## ------------------------------------------------------------
##
## Table: Table continues below
##
##
## ---------------------------------------------------------
##           residuals                   coefficients
## ------------------------------- -------------------------
##  c(‘1‘ = 0.29696704008269, ‘2‘   c(1.70963101019598e-16,
##   = -0.232371604750578, ‘3‘ =     -2.3886124511032e-17,
##    -0.55711075156695, ‘4‘ =       0.198214861974286,
##    -0.244473056416564, ‘5‘ =      0.0329222605594672,
##     1.36487295829493, ‘6‘ =       8.62514038133913e-16,
##   -0.0128600746857614, ‘7‘ =     -7.25531118007121e-16,
##    0.263109273392085, ‘8‘ =       0.999999999999999,
##    0.821687314559134, ‘9‘ =       0.999999999999999)
##    1.06393707947166, ‘10‘ =
##   -0.0768413843602668, ‘11‘ =
##   -0.0351807012835388, ‘12‘ =
##    0.19174339279074, ‘13‘ =
##    0.317210265120091, ‘14‘ =
##   -0.381741856509043, ‘15‘ =
##    1.37630835183472, ‘16‘ =
##    0.373983323033045, ‘17‘ =
##   -0.328848185390352, ‘18‘ =
##    1.15212414740142, ‘19‘ =
##    0.192663066012325, ‘20‘ =
##   -0.912364512935807,  ‘21‘ =
##   -0.494334702137243, ‘22‘ =
##    0.935804320890385, ‘23‘ =
##    0.538030911198243, ‘24‘ =
##    -1.25441526063447, ‘25‘ =
##   -0.449866243029349, ‘26‘ =
##   -0.662520378453251, ‘27‘ =
##    0.478944920042707, ‘28‘ =
##   -0.267918222552493, ‘29‘ =
##    -1.0908938770022, ‘30‘ =
##   -0.749904895499116, ‘31‘ =
##   -0.663740100436094, ‘32‘ =
##   0.0896106663514096, ‘33‘ =
##   -0.705966738882404, ‘34‘ =
##    -3.81640130180478, ‘35‘ =
##    5.01085970904756, ‘36‘ =
##   -0.404503196267755, ‘37‘ =
##    -1.12800455279595, ‘38‘ =
##   0.0709768062113732, ‘39‘ =
##    1.30591390695658,  ‘40‘ =
##   -0.481726375722505, ‘41‘ =
```

```
##   -0.357439957856007, ‘42‘ =
##    0.636810634098707, ‘43‘ =
##   0.0685375255928355, ‘44‘ =
##     3.15419055257569, ‘45‘ =
##     1.81104210983019, ‘46‘ =
##   -0.108478838915796, ‘47‘ =
##    -1.02663153505203, ‘48‘ =
##   -0.893178735416828, ‘49‘ =
##   -0.173543201216286, ‘50‘ =
##   -0.696925040670163, ‘51‘ =
##    -1.01125023119238, ‘52‘ =
##     1.14411071490075, ‘53‘ =
##    -2.80008635291163, ‘54‘ =
##   -0.446552765179383, ‘55‘ =
##     0.14895559747248, ‘56‘ =
##   -0.585199308187902, ‘57‘ =
##   -0.404528502580093, ‘58‘ =
##     -2.2249601848315, ‘59‘ =
##   -2.37644000574289,  ‘60‘ =
##   -0.189057053668257, ‘61‘ =
##    -1.68577656787225, ‘62‘ =
##     0.10044024560532, ‘63‘ =
##    0.222057711014648, ‘64‘ =
##    0.598923910759021, ‘65‘ =
##     0.17423897012573, ‘66‘ =
##    0.557657154395505, ‘67‘ =
##    0.547736377791402, ‘68‘ =
##    0.358053417612559, ‘69‘ =
##    0.192636361960154, ‘70‘ =
##   -0.198460992693402, ‘71‘ =
##   -0.0440316917805067, ‘72‘ =
##    0.163021244627104, ‘73‘ =
##    0.470991634359182, ‘74‘ =
##    0.583290395294567, ‘75‘ =
##  -0.00641220792742011, ‘76‘ =
##   -0.0377992376331464, ‘77‘ =
##     1.01408926088934, ‘78‘ =
##    0.147064302337301, ‘79‘ =
##   -0.332642058432684,  ‘80‘ =
##   -0.301138105117955, ‘81‘ =
##    0.736934791596301, ‘82‘ =
##    0.811505932063739, ‘83‘ =
##   -0.268693870256762, ‘84‘ =
##   0.0438448338746104, ‘85‘ =
##    0.437197702780659, ‘86‘ =
##    0.523328928660125, ‘87‘ =
##    0.287759561640989, ‘88‘ =
##    -1.70378610739052, ‘89‘ =
##   -0.912877731636869, ‘90‘ =
##    0.574084256463748, ‘91‘ =
##   -0.0723137030808667, ‘92‘ =
##    0.104509315040244, ‘93‘ =
##   -0.0228727638860093, ‘94‘ =
##     -0.2673289435028, ‘95‘ =
```

```
##    1.34676823904706, ‘96‘ =
##    0.560539677374614, ‘97‘ =
##    0.970434913326458, ‘98‘ =
##    0.010952329141419, ‘99‘ =
##      -0.246062387194736 )
## ---------------------------------------------------------
##
## Table: Table continues below
##
##
## ----------------------------------------------------------------
##             aliased                  sigma            df
## ------------------------------- ------------------ -------------
##  c(‘(Intercept)‘ = FALSE, yhat   1.07115476917419   c(2, 97, 2)
##            = FALSE)
## ----------------------------------------------------------------
##
## Table: Table continues below
##
##
## ----------------------------------------------------------------------------
##       r.squared           adj.r.squared              fstatistic
## ---------------------- --------------------- -------------------------------
##  2.24885705193877e-32   -0.0103092783505154            c(value =
##                                              2.18139134038061e-30, numdf =
##                                                     1, dendf = 97)
## ----------------------------------------------------------------------------
##
## Table: Table continues below
##
##
## -----------------------
##      cov.unscaled
## -----------------------
##  c(0.0342426981246771,
##  -0.0047755262724861,
##  -0.0047755262724861,
##  0.00094465851587709)
## -----------------------
```

Similarly another BPG test returns a p-value < 0.05 so we reject the null hypothesis of homoskedasticity. We can say our estimator is unbiased but not minimum variance and efficient.

```r
bptest(ur.ardl.model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  ur.ardl.model
## BP = 41.037, df = 6, p-value = 2.847e-07
```

Testing for normality using the Jarque Bera test we see a χ^2 value of 380 and a p-value $<$ 2.2e-16, so we reject the null hypothesis of normality.

```r
jarque.bera.test(u)
```

```
##
##  Jarque Bera Test
##
## data:  u
## X-squared = 150.86, df = 2, p-value < 2.2e-16
```

This implies our estimator is no longer the Maximum Likelihood estimator but is the minimum variance estimator in the class of linear unbiased estimators.

Performing a RESET test for functional form and non-linearity we see a p-value of > 0.05. We fail to reject the null hypothesis correct functional form and linearity. The RESET test takes the form of $y_t = \hat{\beta}_t x_t + \hat{u}_t$ then taking the residuals and $\hat{u}_t = b' x_t + c\hat{y}_t^2 + v_t$

```r
resettest(ur.ardl.model)
```

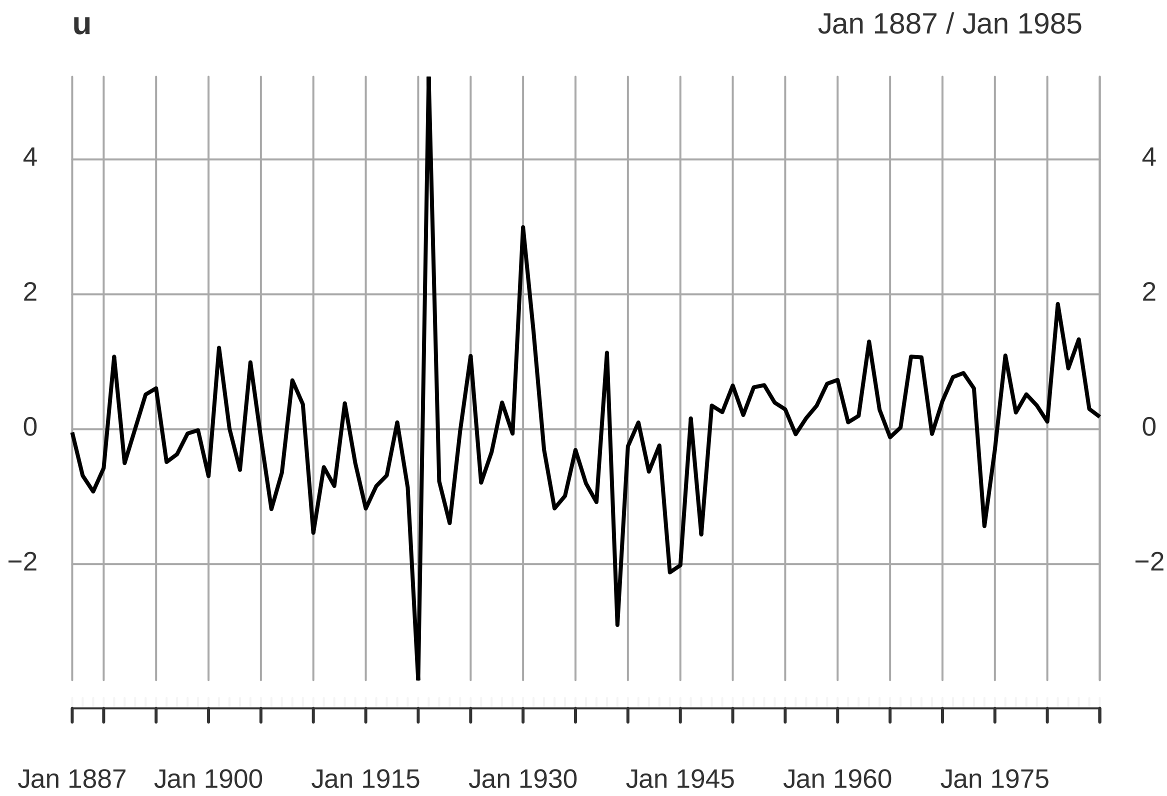
```
##
##  RESET test
##
## data:  ur.ardl.model
## RESET = 0.16425, df1 = 2, df2 = 90, p-value = 0.8488
```

Restricted Model

We consider the model $\Delta U_t = \alpha_0 + \alpha_1 U_{t-1} + b_0 \Delta LQ_t + \epsilon_{2t}$

```r
r.data <- cbind(ur.data$U - ur.data$U.1, ur.data$U.1 - ur.data$U.2, ur.data$LQ - ur.data$LQ.1)

r.model <- dyn$lm(U ~ U.1 + LQ, data = r.data)
u <- resid(r.model)
plot(u)
```




```r
pandoc.table(summary(r.model))
```

```
##
## ---------------------------------------------------------------------------
##              call                   terms                residuals
## ---------------------------- -------------- ------------------------------
##  lm(formula = dyn(U ~ U.1 +   U ~ U.1 + LQ   c(`3` = -0.0472111449972483,
##     LQ), data = r.data)                      `4` = -0.688679659534648, `5`
##                                               = -0.924318758616722, `6` =
##                                                -0.576648714721268, `7` =
##                                                 1.07570828144826, `8` =
##                                                -0.503375989521515, `9` =
##                                               0.00527121412239956, `10` =
##                                                0.514048778719633, `11` =
##                                                0.606826833781547, `12` =
##                                               -0.487096521063751, `13` =
##                                               -0.370692569847783, `14` =
##                                               -0.0638748304086236, `15` =
##                                               -0.0150071675185386, `16` =
##                                               -0.696061259519742, `17` =
##                                                 1.207795739373, `18` =
##                                               0.00858920655175497, `19` =
##                                               -0.603676255910197, `20` =
##                                                0.992414737446694, `21` =
##                                               -0.13137757648756,  `22` =
```

```
##                                                  -1.18531355189013, `23` =
##                                                  -0.64339904480474, `24` =
##                                                  0.725244930449776, `25` =
##                                                   0.3690593744814, `26` =
##                                                  -1.53691638536141, `27` =
##                                                 -0.562259956512869, `28` =
##                                                  -0.8424670594657, `29` =
##                                                  0.384941168847499, `30` =
##                                                 -0.504024047892761, `31` =
##                                                  -1.17648084493029, `32` =
##                                                 -0.843561268096782, `33` =
##                                                 -0.685088355075968, `34` =
##                                                  0.101838371632778, `35` =
##                                                 -0.864446877311847, `36` =
##                                                  -3.72065294985627, `37` =
##                                                   5.2257518142803, `38` =
##                                                 -0.773170798215132, `39` =
##                                                  -1.39280096917047, `40` =
##                                                 -0.0289397150970434, `41` =
##                                                  1.08748597873752,  `42` =
##                                                 -0.792877676760683, `43` =
##                                                 -0.339902119892234, `44` =
##                                                  0.395349254697341, `45` =
##                                                 -0.0661010697249275, `46` =
##                                                  2.99610329484568, `47` =
##                                                  1.44684620323022, `48` =
##                                                 -0.300442346937601, `49` =
##                                                  -1.17463181622339, `50` =
##                                                 -0.989592390658515, `51` =
##                                                 -0.307162641106619, `52` =
##                                                 -0.803157003755422, `53` =
##                                                  -1.08125717013338, `54` =
##                                                  1.13452995863472, `55` =
##                                                  -2.90343382380995, `56` =
##                                                 -0.253749578144499, `57` =
##                                                  0.100202612843317, `58` =
##                                                 -0.629635661702343, `59` =
##                                                 -0.240833926944963, `60` =
##                                                 -2.12299775262165,  `61` =
##                                                  -2.0176196554007, `62` =
##                                                  0.160484946094672, `63` =
##                                                  -1.56202804082592, `64` =
##                                                  0.350198827579442, `65` =
##                                                  0.252887014880316, `66` =
##                                                  0.647646487911608, `67` =
##                                                  0.210157809518285, `68` =
##                                                   0.6211793531989, `69` =
##                                                  0.653251225681689, `70` =
##                                                  0.393871629350217, `71` =
##                                                  0.294595589338634, `72` =
##                                                 -0.074501611471445, `73` =
##                                                  0.163189151224726, `74` =
##                                                  0.34823593002963, `75` =
##                                                  0.674595128174591, `76` =
```

```
##                                              0.730958117124665, `77` =
##                                              0.102186234374408, `78` =
##                                              0.197801291002285, `79` =
##                                               1.2997129686036, `80` =
##                                             0.289039400689105,  `81` =
##                                             -0.120077871112797, `82` =
##                                             0.0247644098392756, `83` =
##                                              1.07576226008971, `84` =
##                                              1.06611840968791, `85` =
##                                             -0.0704844132510414, `86` =
##                                              0.41775386742514, `87` =
##                                              0.772822774990656, `88` =
##                                               0.8327073034129, `89` =
##                                              0.605411329092948, `90` =
##                                              -1.43735845958505, `91` =
##                                             -0.301143023785955, `92` =
##                                              1.09266750418575, `93` =
##                                              0.246989581335989, `94` =
##                                              0.517232351691401, `95` =
##                                              0.351331488860759, `96` =
##                                              0.111011745138688, `97` =
##                                              1.85759019827775, `98` =
##                                              0.900939721053683, `99` =
##                                              1.3326596466999, `100` =
##                                             0.300199729351747,  `101` =
##                                                0.182569145643267)
## --------------------------------------------------------------------------
##
## Table: Table continues below
##
##
## --------------------------------------------------------------------------
##      coefficients                 aliased                    sigma
## ----------------------- ------------------------------- ------------------
##  c(0.677498325218963,    c(`(Intercept)` = FALSE, U.1 =   1.10997687099591
##   0.041147817102772,           FALSE, LQ = FALSE)
##   -31.4235545880268,
##   0.132216641440925,
##   0.0751892491772549,
##    3.49278438078747,
##    5.12415319157591,
##   0.547256656410653,
##   -8.99670611242891,
##  1.54397348475018e-06,
##    0.58547241012766,
##  2.12657118492354e-14)
## --------------------------------------------------------------------------
##
## Table: Table continues below
##
##
## ------------------------------------------------------
##      df            r.squared         adj.r.squared
## ------------- ------------------- -------------------
```

```
##  c(3, 96, 3)   0.465541845705802   0.454407300824673
## ----------------------------------------------------
##
## Table: Table continues below
##
##
## -----------------------------------------------------------------
##          fstatistic                cov.unscaled        na.action
## ---------------------------- ------------------------ -----------
##  c(value = 41.8105859445413,   c(0.0141887580616992,       1:2
##    numdf = 2, dendf = 96)      -0.000740930572938802,
##                                 -0.200917178177524,
##                                -0.000740930572938802,
##                                 0.0045886363114114,
##                                  0.025544446204396,
##                                 -0.200917178177524,
##                                  0.025544446204396,
##                                  9.90183519913851)
## -----------------------------------------------------------------
```

Fitting the restricted model we see that a level change in log GDP is significant at the 1% level, the change in unemployment is not significant. Plotting the residuals we see similar issues with the post-war period showing a different, possibly more constant variance. We see the same issues around the interwar years with large outliers and higher variance.

Again we no serial correlation with a DW statistic of approximately 2 (2.05).

```r
durbinWatsonTest(as.vector(u))
```

```
## [1] 2.050358
```

Testing again for We test for heteroskedasticity using $\hat{u}_t^2 = \alpha + b'z_t + v_t$ using the hypothesis that $b' = 0$. In this case we use $z_t = x_t$

```r
u2 <- u * u
summary(dyn$lm(u2 ~  U.1 + LQ, data = r.data))$fstatistic
```

```
##    value    numdf    dendf
## 15.92011  2.00000 96.00000
```

We see that the F-statstic is 9.5, suggesting the we fail to accept the null hypothesis of $b' = 0$ and of homoskedasticity. Similary a standalone BPG test shows we cannot accept the null hypothesis of constant variance

```r
bptest(r.model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  r.model
## BP = 24.657, df = 2, p-value = 4.423e-06
```

Testing for normality again we see that we fail to accept the null hypothesis of normality using the Jarque Bera Test

```
jarque.bera.test(u)
```

```
##
##  Jarque Bera Test
##
## data:  u
## X-squared = 129.16, df = 2, p-value < 2.2e-16
```

Performing a RESET test to inform us on the functional form we see a RESET test statistic of 4.9 and a p-value of 0.009. We reject the null hyphothesis of non-linearity or correct functional form

```
resettest(r.model)
```

```
##
##  RESET test
##
## data:  r.model
## RESET = 7.1321, df1 = 2, df2 = 94, p-value = 0.001307
```

Comparing the tests between the two models we see little difference bar the result of the RESET test which in the case of the restricted model fails for functional form. We observe heterskedasticity in both as well as non normality of the residuals. We do not observe any effects of serial correlation.

Comparing the AIC & BIC tests we see that AIC & BIC prefers the restricted model marginally.

```
data.frame(AIC(ur.ardl.model), AIC(r.model), BIC(ur.ardl.model), BIC(r.model))
```

```
##   AIC.ur.ardl.model. AIC.r.model. BIC.ur.ardl.model. BIC.r.model.
## 1           308.5393     306.5626           329.3003     316.9431
```

Looking at the likelihood ratio test $2(LL(\theta) - LL(\theta^*))$ we see

```
2 * (logLik(ur.ardl.model) - logLik(r.model))
```

```
## 'log Lik.' 6.023258 (df=8)
```

We have 7 parameters in the unrestricted model and 3 parameters in the restricted so we have imposed 4 restrictions. We have restricted the following coefficients $\alpha_2 = (\alpha_1 - 1); \beta_1 = -\beta_0; \beta_2 = 0; \gamma = 0$ with the remaining coefficients unchanged.

Using the likelihood ratio test (which is $\chi^2(4)$ distributed) we fail to reject the null hypothesis that the restrictions are not binding, or more simply we can not say the restrictions are binding.

ARIMA

We test the following variables, U_t, RL_t , G_t and INF_t for unit roots. Testing each for unit roots with and without trends we see the following

- U_t: We fail to reject the null hypothesis that unemployment has a unit root and is of at least I(1)
- RL_t: We fail to reject the null hypothesis that unemployment has a unit root and is of at least I(1)
- INF_t: Inflation does not have a unit root
- G_t: Growth does not have a unit root

```r
summary(ur.df(macro.series$INF))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.6013  -0.7976   0.5862   2.1137  13.4810
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.18818    0.05798  -3.246  0.00161 **
## z.diff.lag  0.15017    0.10063   1.492  0.13885
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.96 on 97 degrees of freedom
## Multiple R-squared:  0.1007, Adjusted R-squared:  0.08213
## F-statistic: 5.429 on 2 and 97 DF,  p-value: 0.005821
##
##
## Value of test-statistic is: -3.2457
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```r
summary(ur.df(macro.series$G))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.966 -0.755  1.275  2.632 10.640
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.41487    0.09906  -4.188  6.2e-05 ***
## z.diff.lag -0.12021    0.10133  -1.186    0.238
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.243 on 97 degrees of freedom
## Multiple R-squared:  0.2443, Adjusted R-squared:  0.2287
## F-statistic: 15.68 on 2 and 97 DF,  p-value: 1.261e-06
##
##
## Value of test-statistic is: -4.1881
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```r
summary(ur.df(macro.series$U))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4967 -0.4548  0.0718  0.5684  9.2695
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.02529    0.02519  -1.004    0.318
## z.diff.lag  0.14093    0.10184   1.384    0.170
##
## Residual standard error: 1.492 on 97 degrees of freedom
## Multiple R-squared:  0.02527,    Adjusted R-squared:  0.005168
## F-statistic: 1.257 on 2 and 97 DF,  p-value: 0.2891
##
##
## Value of test-statistic is: -1.0038
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```r
summary(ur.df(macro.series$U, type="trend"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4097 -0.6137 -0.2345  0.4794  8.9620
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.377221   0.396819   0.951   0.3442
## z.lag.1     -0.099829   0.045615  -2.189   0.0311 *
## tt           0.003140   0.005237   0.600   0.5502
## z.diff.lag   0.175586   0.103506   1.696   0.0931 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.473 on 95 degrees of freedom
## Multiple R-squared:  0.06874,    Adjusted R-squared:  0.03933
## F-statistic: 2.337 on 3 and 95 DF,  p-value: 0.07853
##
##
## Value of test-statistic is: -2.1885 1.8541 2.7449
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

```r
summary(ur.df(macro.series$RL))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
```

```
## Residuals:
##      Min       1Q  Median      3Q     Max
## -1.9742 -0.1174  0.0296  0.1992  3.7896
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1     0.00251    0.01077   0.233   0.8162
## z.diff.lag  0.26079    0.09952   2.620   0.0102 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6253 on 97 degrees of freedom
## Multiple R-squared:  0.07072,    Adjusted R-squared:  0.05156
## F-statistic: 3.691 on 2 and 97 DF,  p-value: 0.02851
##
##
## Value of test-statistic is: 0.2331
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```r
summary(ur.df(macro.series$RL, type="trend"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q  Median      3Q     Max
## -1.6500 -0.2288  0.0274  0.1283  3.8623
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.041577   0.128312   0.324  0.74663
## z.lag.1     -0.066426   0.030854  -2.153  0.03386 *
## tt           0.006769   0.003361   2.014  0.04682 *
## z.diff.lag   0.281447   0.098117   2.868  0.00508 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6138 on 95 degrees of freedom
## Multiple R-squared:  0.1115, Adjusted R-squared:  0.08346
## F-statistic: 3.974 on 3 and 95 DF,  p-value: 0.01025
##
##
## Value of test-statistic is: -2.1529 1.9103 2.4806
```

```
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

We estimate a random walk (using OLS) for inflation of the following form $y_t = \alpha + y_{t-1} + \epsilon_t$

```r
rw.model <- dyn$lm(INF ~ stats::lag(INF, k=1), data=macro.series)
pandoc.table(summary(rw.model))
```

```
##
## ---------------------------------------------------------------
##              call                           terms
## ------------------------------- -------------------------------
##     lm(formula = dyn(INF ~       INF ~ stats::lag(INF, k = 1)
##  stats::lag(INF, k = 1)), data
##         = macro.series)
## ---------------------------------------------------------------
##
## Table: Table continues below
##
##
## ---------------------------------------------------------
##            residuals                   coefficients
## -------------------------------- ----------------------
##  c(‘2‘ = 0.290949321220401, ‘3‘   c(0.933071632274192,
##    = -1.48781077591828, ‘4‘ =      0.768288048752292,
##    -0.749809581822943, ‘5‘ =       0.458901022101705,
##     1.06252530003253, ‘6‘ =        0.0642420062228157,
##     -1.08688392186767, ‘7‘ =        2.03327425160408,
##     -5.1977051495614, ‘8‘ =         11.9592785768174,
##     1.92214932588614, ‘9‘ =        0.0447285975112509,
##    -0.201286193546137, ‘10‘ =     7.32335715065377e-21)
##    -1.12476322053219, ‘11‘ =
##    -2.55470778678776, ‘12‘ =
##    -0.629691652087585, ‘13‘ =
##    0.595131587847864, ‘14‘ =
##    -1.05141482418085, ‘15‘ =
##    0.853579753901027, ‘16‘ =
##     1.12262070174393, ‘17‘ =
##    -5.29176166971837, ‘18‘ =
##    -0.949520099192978, ‘19‘ =
##    -0.384542709815678, ‘20‘ =
##    -1.77922059992403, ‘21‘ =
##    0.295041565787919,  ‘22‘ =
##    0.240226376485795, ‘23‘ =
##    -0.92699426620397, ‘24‘ =
##    -2.65655573938288, ‘25‘ =
##    -0.514774206073703, ‘26‘ =
##    0.625586435454085, ‘27‘ =
##     -1.4328323662627, ‘28‘ =
```

```
##     1.08757022112139, ‘29‘ =
##     -2.8765566403602, ‘30‘ =
##    -1.69569148410718, ‘31‘ =
##     12.4003066547042, ‘32‘ =
##     2.92894188859642, ‘33‘ =
##     10.4548691402764, ‘34‘ =
##    -2.47879812542358, ‘35‘ =
##     2.6528063886837, ‘36‘ =
##     2.32810371954462, ‘37‘ =
##    -21.9232635528816, ‘38‘ =
##    -8.41383923437332, ‘39‘ =
##     2.80816434710761, ‘40‘ =
##     2.86860057518438, ‘41‘ =
##    0.366800010469724,  ‘42‘ =
##    -1.15347220282135, ‘43‘ =
##    -2.42639028043812, ‘44‘ =
##    -0.423289979511669, ‘45‘ =
##    -1.07774166675046, ‘46‘ =
##    -1.15227502948317, ‘47‘ =
##    -2.20610288800667, ‘48‘ =
##    -1.70281611910472, ‘49‘ =
##    -0.563522226155314, ‘50‘ =
##    -0.717659479123523, ‘51‘ =
##    0.460843575196156, ‘52‘ =
##    -0.662537725925388, ‘53‘ =
##     1.69553153234098, ‘54‘ =
##    -1.14118614587273, ‘55‘ =
##    0.284557871576798, ‘56‘ =
##     6.68164958993729, ‘57‘ =
##     1.74043148396042, ‘58‘ =
##    -1.47865489480177, ‘59‘ =
##    -2.05625905831363, ‘60‘ =
##    0.947497930750282, ‘61‘ =
##    -3.0884671991185,  ‘62‘ =
##    0.658371634588932, ‘63‘ =
##     5.73390104051095, ‘64‘ =
##    -1.66985057868732, ‘65‘ =
##    -2.66138684957842, ‘66‘ =
##    -1.96304184357188, ‘67‘ =
##     5.97362302760734, ‘68‘ =
##    -0.322408487015252, ‘69‘ =
##    -4.44427813806179, ‘70‘ =
##    -1.03604161656282, ‘71‘ =
##     2.79295039141944, ‘72‘ =
##     2.15472292238203, ‘73‘ =
##    -2.05953064543092, ‘74‘ =
##    -0.304906483125652, ‘75‘ =
##    -3.01561873911594, ‘76‘ =
##    -0.472964910205928, ‘77‘ =
##     1.72002442950144, ‘78‘ =
##     -0.1237580475311, ‘79‘ =
##    -2.22671665954438, ‘80‘ =
##     1.50532051500614, ‘81‘ =
##    2.11214807348203,  ‘82‘ =
```

```
##    -0.285052649324036, ‘83‘ =
##    -1.86981843850848, ‘84‘ =
##    0.834200253475052, ‘85‘ =
##     2.18736697324145, ‘86‘ =
##     3.43425422419681, ‘87‘ =
##   -0.0204601465583053, ‘88‘ =
##    0.309500573673274, ‘89‘ =
##     1.85134619019903, ‘90‘ =
##     7.50493316570062, ‘91‘ =
##     10.7798522552487, ‘92‘ =
##    -4.31371565612192, ‘93‘ =
##    0.919717573853207, ‘94‘ =
##    0.243823826851448, ‘95‘ =
##     4.04833127849409, ‘96‘ =
##     7.03650306209312, ‘97‘ =
##     -3.4644513362072, ‘98‘ =
##    -2.26748687785294, ‘99‘ =
##    -1.3649708276438, ‘100‘ =
##   -0.0432438791892962, ‘101‘ =
##       0.673124795954207 )
## --------------------------------------------------------
##
## Table: Table continues below
##
##
## -------------------------------------------------------------
##           aliased                  sigma             df
## ---------------------------- ------------------ -------------
##   c(‘(Intercept)‘ = FALSE,    3.90565926392268   c(2, 98, 2)
##  ‘stats::lag(INF, k = 1)‘ =
##            FALSE)
## -------------------------------------------------------------
##
## Table: Table continues below
##
##
## -------------------------------------------------------------------
##      r.squared        adj.r.squared            fstatistic
## ------------------- ----------------- -----------------------------
##  0.593402067434665   0.5892531089391   c(value = 143.024344077924,
##                                          numdf = 1, dendf = 98)
## -------------------------------------------------------------------
##
## Table: Table continues below
##
##
## ------------------------------------
##      cov.unscaled        na.action
## ----------------------- ------------
##  c(0.0138054111707068,   c(‘1‘ = 1)
##  -0.0010146716759569,
##  -0.0010146716759569,
##  0.000270551213470566)
## ------------------------------------
```

We estimate the random walk and the ARMA using Maximum Likelihood

```r
ma.model <- arima(macro.series[, "INF"], order = c(1, 0, 1), method="ML")
rw.model <- arima(macro.series[, "INF"], order = c(1, 0, 0), method="ML")
pandoc.table(summary(ma.model))
```

```
##
## --------------------------------------------
##             Length   Class      Mode
## --------------- -------- -------- -----------
##    **coef**        3      -none-    numeric
##
##   **sigma2**       1      -none-    numeric
##
##  **var.coef**      9      -none-    numeric
##
##    **mask**        3      -none-    logical
##
##   **loglik**       1      -none-    numeric
##
##     **aic**        1      -none-    numeric
##
##    **arma**        7      -none-    numeric
##
##  **residuals**    101       ts      numeric
##
##    **call**        4      -none-     call
##
##   **series**       1      -none-   character
##
##    **code**        1      -none-    numeric
##
##   **n.cond**       1      -none-    numeric
##
##    **nobs**        1      -none-    numeric
##
##    **model**       10     -none-     list
## --------------------------------------------
```

```r
pandoc.table(summary(rw.model))
```

```
##
## --------------------------------------------
##             Length   Class      Mode
## --------------- -------- -------- -----------
##    **coef**        2      -none-    numeric
##
##   **sigma2**       1      -none-    numeric
##
##  **var.coef**      4      -none-    numeric
##
##    **mask**        2      -none-    logical
##
```

```
##    **loglik**       1      -none-    numeric
##
##      **aic**        1      -none-    numeric
##
##     **arma**        7      -none-    numeric
##
##   **residuals**    101       ts      numeric
##
##     **call**        4      -none-     call
##
##    **series**       1      -none-   character
##
##     **code**        1      -none-    numeric
##
##    **n.cond**       1      -none-    numeric
##
##     **nobs**        1      -none-    numeric
##
##     **model**       10     -none-     list
## ---------------------------------------------
```

```
ma.model$aic
```

```
## [1] 563.557
```

```
rw.model$aic
```

```
## [1] 566.3142
```

```
2 * (ma.model$loglik - rw.model$loglik)
```

```
## [1] 4.757261
```

```
qchisq(0.025, df=1, lower.tail = F)
```

```
## [1] 5.023886
```

Using AIC we marginally choose the ARMA model over the random walk.