Computer Lab Session 3:
Multiple Regression Model in R
EF3450 Semester B 2017-18

The example below is similar to problem in hand-in assignment 3, but uses a different data set. This handout explains the relevant R commands and output.

Uncovered Interest Parity (UIP)

The interest parity condition

$$1.(1+r_t) = (E[S_{t+1}]/S_t) * (1+r_t^*)$$

have a common approximation as

$$2.E[S_{t+1}] - S_t/S_t = r_t - r_t^*$$

which can be tested by considering the regression model as:

$$3.E[S_{t+1}] - S_t/S_t = \beta_0 + \beta_1 * r_t + \beta_2 * r_t^* + e_t$$

`s_Aus` : S_t, (the spot exchange rate, expressed as the price in AUD for one USD)

`s_ch_Aus`: $E[S_{t+1}] - S_t/S_t$ `* 100`, (change in the spot exchange rate, percentage)

`r_Aus` : r_t (the return of a 3-month Australia treasury bill, %)

`r_US` : r_t^* (the return of a 3-month US treasury bill, %)

Package

car is required:

```
install.packages('car')

library(car)
```

Load data into R

```
setwd('C:\\Users\\EFUser\\Desktop')
data_CLS3 <- read.csv('UIP_dataset_5.csv', stringsAsFactors = F)
str(data_CLS3)
## 'data.frame':    130 obs. of  5 variables:
##  $ Date    : chr  "3/31/1970" "6/30/1970" "9/30/1970" "12/31/1970" ...
##  $ s_Aus   : num  0.89 0.89 0.89 0.89 0.89 0.89 0.89 0.86 0.84 0.84 ...
##  $ s_ch_Aus: num  0 0 0 0 0 ...
##  $ r_Aus   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ r_US    : num  1.768 1.646 1.556 1.314 0.954 ...
```

Date transformation

```
data_CLS3 <- transform(data_CLS3, Date = as.Date(Date, "%m/%d/%Y") )
str(data_CLS3); head(data_CLS3,2); tail(data_CLS3,2)
```

```
## 'data.frame':    130 obs. of  5 variables:
##  $ Date    : Date, format: "1970-03-31" "1970-06-30" ...
##  $ s_Aus   : num  0.89 0.89 0.89 0.89 0.89 0.89 0.89 0.86 0.84 0.84 ...
##  $ s_ch_Aus: num  0 0 0 0 0 ...
##  $ r_Aus   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ r_US    : num  1.768 1.646 1.556 1.314 0.954 ...
##         Date s_Aus s_ch_Aus r_Aus     r_US
## 1 1970-03-31  0.89        0    NA 1.767580
## 2 1970-06-30  0.89        0    NA 1.646393
##           Date s_Aus  s_ch_Aus    r_Aus      r_US
## 129 2002-03-31  1.93 -6.217617 1.041128 0.4297221
## 130 2002-06-30  1.81  1.104972 1.147594 0.4272540
```

Regression for UIP

```
reg_UIP <- lm( s_ch_Aus ~ r_Aus + r_US, data = data_CLS3)
```

Regression Result

```
summary(reg_UIP)
```

```
##
## Call:
## lm(formula = s_ch_Aus ~ r_Aus + r_US, data = data_CLS3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7303 -2.4788 -0.3899  2.1063 14.8365
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.5048     1.2180  -0.414   0.6796
## r_Aus        -0.9742     0.6104  -1.596   0.1143
## r_US          2.4141     0.9360   2.579   0.0117 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.134 on 83 degrees of freedom
##   (44 observations deleted due to missingness)
## Multiple R-squared:  0.07473,    Adjusted R-squared:  0.05243
## F-statistic: 3.352 on 2 and 83 DF,  p-value: 0.03983
```

Report all coefficient estimates

```
reg_summary_UIP <- summary(reg_UIP)
reg_summary_UIP$coefficients[1,1]
## [1] -0.5048412
reg_summary_UIP$coefficients[2,1]
## [1] -0.9742184
reg_summary_UIP$coefficients[3,1]
## [1] 2.414129
```

95% confidence Interval for beta_1 and beta_2

`confint()` computes confidence intervals for one or more parameters in a fitted model. There is a default and a method for objects inheriting from class "lm".

```
confint(reg_UIP, level=0.95)
```

```
##                  2.5 %    97.5 %
## (Intercept) -2.9274544 1.9177719
## r_Aus       -2.1881920 0.2397552
## r_US         0.5525206 4.2757377
```

```
Alternatively, if you would like only selected parameter
confint(object, parm, level = 0.95, ...)
```

```
#confint(reg_UIP, 'r_Aus') # 95%CI for beta_1
#confint(reg_UIP, 'r_US') # 95%CI for beta_2
```

Report the goodness of fit

```
reg_summary_UIP$r.squared
## [1] 0.07472651
reg_summary_UIP$adj.r.squared
## [1] 0.05243077
```

Hypothesis Testing

H0: Beta_0 <= 0 (one-tail right test)

To test restriction on one parameter we can still calculate t-statstics and then calculate p-value

```
t_stat_intercept <- ( reg_summary_UIP$coefficients[1,1]- 0 )/
reg_summary_UIP$coefficients[1,2]
pt(t_stat_intercept, df=83, lower.tail = FALSE)
## [1] 0.6602017
pt(t_stat_intercept, df=83, lower.tail = FALSE) < 0.05
## [1] FALSE
#Cannot reject the H0
pnorm(t_stat_intercept, lower.tail = FALSE) #approximately normal by large
sample
## [1] 0.6607363
pnorm(t_stat_intercept, lower.tail = FALSE) < 0.05
## [1] FALSE
#Cannot reject the H0
```

But for multiple parameters or restrictions we need linearHypothesis() from car package. Let's try function from one restriction on one parameter case

```
linearHypothesis(reg_UIP, "(Intercept)=0")
## Linear hypothesis test
##
## Hypothesis:
## (Intercept) = 0
```

lm object

Each restruction is wrap by “ ”
LS: regressor(s)
RS: hypothesized value

```
##
## Model 1: restricted model
## Model 2: s_ch_Aus ~ r_Aus + r_US
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1      84 1421.6
## 2      83 1418.6  1    2.9362 0.1718 0.6796
p_value_both <- linearHypothesis(reg_UIP, "(Intercept)=0")[2,6]
1 - p_value_both/2
## [1] 0.6602017
# 1. by symmetric right tail = left tail hence divided by 2,
# and 2. P(X<=t) = 1- P(X>=t)
1 - p_value_both/2 < 0.05
## [1] FALSE
```
Cannot reject the H0

p-value for the two tailed test
- If you want only one side shaded area divided it by two

H0: Beta_1 = 1

```
linearHypothesis(reg_UIP, "r_Aus = 1")
## Linear hypothesis test
##
## Hypothesis:
## r_Aus = 1
##
## Model 1: restricted model
## Model 2: s_ch_Aus ~ r_Aus + r_US
##
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1      84 1597.5
## 2      83 1418.6  1    178.82 10.462 0.00175 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
linearHypothesis(reg_UIP, "r_Aus = 1")[2, 6] <0.05
## [1] TRUE
```
Reject the H0

H0: Beta_1 = -1

```
linearHypothesis(reg_UIP, "r_Aus = -1")
## Linear hypothesis test
##
## Hypothesis:
## r_Aus = - 1
##
## Model 1: restricted model
## Model 2: s_ch_Aus ~ r_Aus + r_US
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1      84 1418.7
## 2      83 1418.6  1  0.030496 0.0018 0.9664
linearHypothesis(reg_UIP, "r_Aus = -1")[2, 6] <0.05
## [1] FALSE
```
Cannot reject the H0

H0: Beta_1 + Beta_2 = 0

```
linearHypothesis(reg_UIP, "r_Aus + r_US = 0")
## Linear hypothesis test
##
## Hypothesis:
## r_Aus  + r_US = 0
##
## Model 1: restricted model
## Model 2: s_ch_Aus ~ r_Aus + r_US
##
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     84 1494.2
## 2     83 1418.6  1    75.601 4.4232 0.03848 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
linearHypothesis(reg_UIP, "r_Aus + r_US = 0")[2, 6] <0.05
## [1] TRUE
```

Reject the H0 and conclude Beta_1 and Beta_2 cannot offset each other

Restriction on two parameter

H0: Intercept = 0, Beta_1 = -1, Beta_2 = 1

```
linearHypothesis(reg_UIP, c("(Intercept) = 0", "r_Aus = -1", "r_US = 1"))
## Linear hypothesis test
##
## Hypothesis:
## (Intercept) = 0
## r_Aus = - 1
## r_US = 1
##
## Model 1: restricted model
## Model 2: s_ch_Aus ~ r_Aus + r_US
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     86 1751.9
## 2     83 1418.6  3    333.29 6.4999 0.0005273 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
linearHypothesis(reg_UIP, c("(Intercept) = 0", "r_Aus = -1", "r_US = 1"))[2,
6] <0.05
## [1] TRUE
```

Reject the H0: either Intercept is significantly different from 0, or Beta_1 is significantly is different from 1, or Beta_2 is significantly different from -1, or both (or all)

Multiple restrictions are placed within c( ) and separated by ,

Test whether lag exchange rate changes have effect on exchange rate changes

1. Creage a lag variable for exchage rate changes

```
data_CLS3$s_ch_Aus_1lag <-c(NA, data_CLS3$s_ch_Aus[-nrow(data_CLS3)])
head(data_CLS3,3); tail(data_CLS3,3)
##         Date  s_Aus s_ch_Aus r_Aus     r_US s_ch_Aus_1lag
## 1 1970-03-31  0.89        0    NA 1.767580            NA
## 2 1970-06-30  0.89        0    NA 1.646393             0
## 3 1970-09-30  0.89        0    NA 1.555814             0
##            Date s_Aus  s_ch_Aus    r_Aus      r_US s_ch_Aus_1lag
## 128 2001-12-31  1.95 -1.025641 1.031432 0.4864391      0.000000
## 129 2002-03-31  1.93 -6.217617 1.041128 0.4297221     -1.025641
## 130 2002-06-30  1.81  1.104972 1.147594 0.4272540     -6.217617
```

2. Include the lag term as regressor

```
reg_carryOver <- lm( s_ch_Aus ~ r_Aus + r_US + s_ch_Aus_1lag, data =
data_CLS3)
summary(reg_carryOver)
##
## Call:
## lm(formula = s_ch_Aus ~ r_Aus + r_US + s_ch_Aus_1lag, data = data_CLS3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.3874 -2.5605 -0.5374  2.3686 15.1651
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.4490     1.2171  -0.369   0.7132
## r_Aus          -0.9264     0.6108  -1.517   0.1332
## r_US            2.2302     0.9486   2.351   0.0211 *
## s_ch_Aus_1lag   0.1209     0.1072   1.127   0.2630
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.128 on 82 degrees of freedom
##   (44 observations deleted due to missingness)
## Multiple R-squared:  0.08884,    Adjusted R-squared:  0.05551
## F-statistic: 2.665 on 3 and 82 DF,  p-value: 0.05323
```

3. Is the carry-over impact significance?

```
reg_carryOver_lagY <- summary(reg_carryOver)$coefficient[4, ]
reg_carryOver_lagY[[4]] < 0.05
## [1] FALSE
Not statistically different from 0 at 5% level
```

Plot for the complete sample period

1. Subset data that have complete observation

`complete.cases`() return a logical vector (TRUE, FALSE, or NA) indicating which cases are complete (in other word, no missing values for a particular row)

```
ix_complete <- complete.cases(data_CLS3)
data_CLS3_complete <- data_CLS3[ix_complete, ]
```

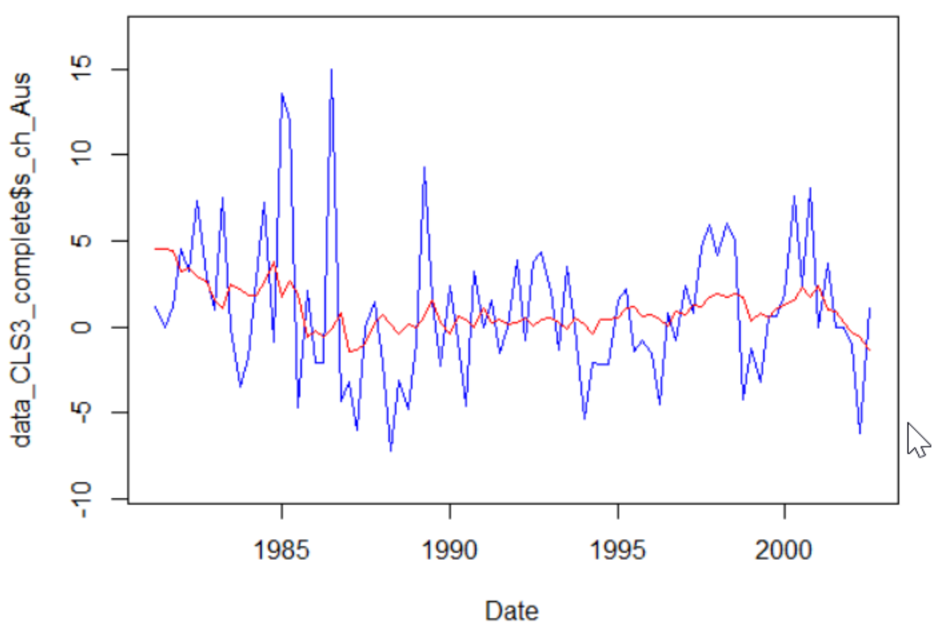
2. Plot the Actual change in exchange rate

```
par(mar = c(5,5,2,5))

plot(data_CLS3_complete$Date, data_CLS3_complete$s_ch_Aus, xlab="Date",
type="l", col="blue",
     cex=0.2, ylim = range(c(min(data_CLS3_complete$s_ch_Aus, na.rm = T)-2,
                             max(data_CLS3_complete$s_ch_Aus, na.rm = T)+2)
),
     xlim = range(min(data_CLS3_complete$Date),
                  max(data_CLS3_complete$Date) ) )
```

3. Add the predicted change in exchange rate on top

```
par(new = T)
plot(data_CLS3_complete$Date, predict(reg_carryOver), type="l", col="red",
     axes=F, xlab="", ylab="", cex=0.2,
     ylim = range(c(min(data_CLS3_complete$s_ch_Aus, na.rm = T)-2,
                    max(data_CLS3_complete$s_ch_Aus, na.rm = T)+2) ) )
```



Plot for sample period between 1980 and 1999

1. Subset data that between 1980 and 1999

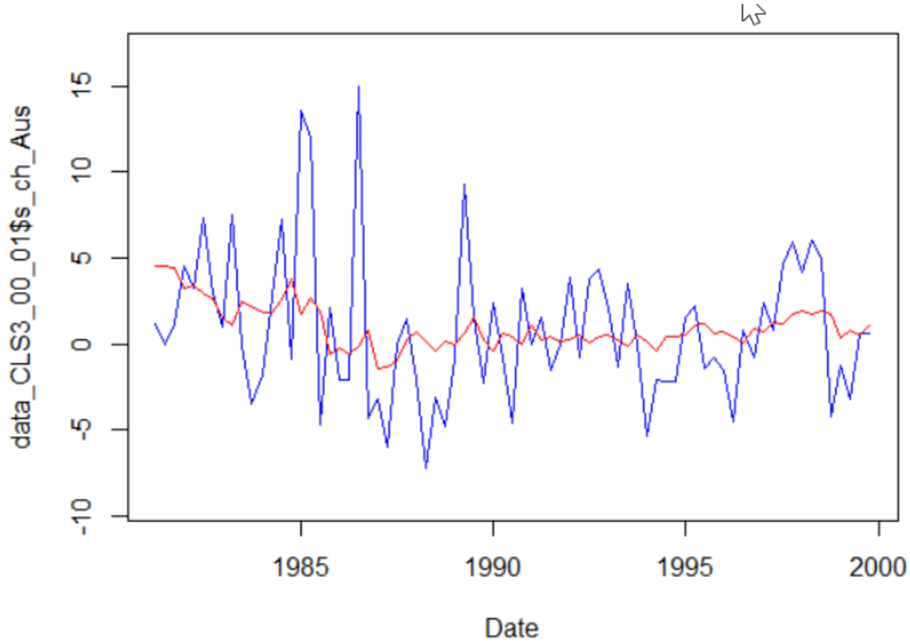
```
ix_80_99 <- (data_CLS3_complete$Date > '1980-01-01' & data_CLS3_complete$Date
< '1999-12-31')
data_CLS3_00_01 <- data_CLS3_complete[ix_80_99, ]
```

2. Plot the Actual change in exchange rate

```
par(mar = c(5,5,2,5))
plot(data_CLS3_00_01$Date, data_CLS3_00_01$s_ch_Aus, xlab="Date", type="l",
col="blue",
     cex=0.2, ylim = range(c(min(data_CLS3_00_01$s_ch_Aus, na.rm = T)-2,
                             max(data_CLS3_00_01$s_ch_Aus, na.rm = T)+2) ),
     xlim = range(min(data_CLS3_00_01$Date),
                  max(data_CLS3_00_01$Date) ) )
```

3. Add the predicted change in exchange rate on top

```
par(new = T)
plot(data_CLS3_00_01$Date, predict(reg_carryOver)[ix_80_99], type="l",
col="red",
     axes=F, xlab="", ylab="", cex=0.2,
     ylim = range(c(min(data_CLS3_00_01$s_ch_Aus, na.rm = T)-2,
                    max(data_CLS3_00_01$s_ch_Aus, na.rm = T)+2) ) )
```



Plot residual against whole sample period

```
plot(data_CLS3$Date[ix_complete], unname(resid(reg_carryOver)) )
lines(data_CLS3$Date[ix_complete], unname(resid(reg_carryOver)), col = 'red')
```