Advanced Financial Statistics - Marked Assignment

Parham Allboje, Arthur Revil, Kelson Ho, Stefan Markovic

11 March, 2021

Question 1

```
us_macro_quarterly1 <- read_excel("data/us_macro_quarterly1.xlsx")
data=us_macro_quarterly1
head(data)
```

```
## # A tibble: 6 x 5
##   freq                GDPC1 JAPAN_IP  PCED   CPI
##   <dttm>              <dbl>    <dbl> <dbl> <dbl>
## 1 1955-01-01 00:00:00 2684.       NA  15.8  26.8
## 2 1955-04-01 00:00:00 2727.       NA  15.8  26.8
## 3 1955-07-01 00:00:00 2764.       NA  15.8  26.8
## 4 1955-10-01 00:00:00 2781.       NA  15.9  26.9
## 5 1956-01-01 00:00:00 2770.       NA  15.9  26.9
## 6 1956-04-01 00:00:00 2793.       NA  16.1  27.0
```

(a)

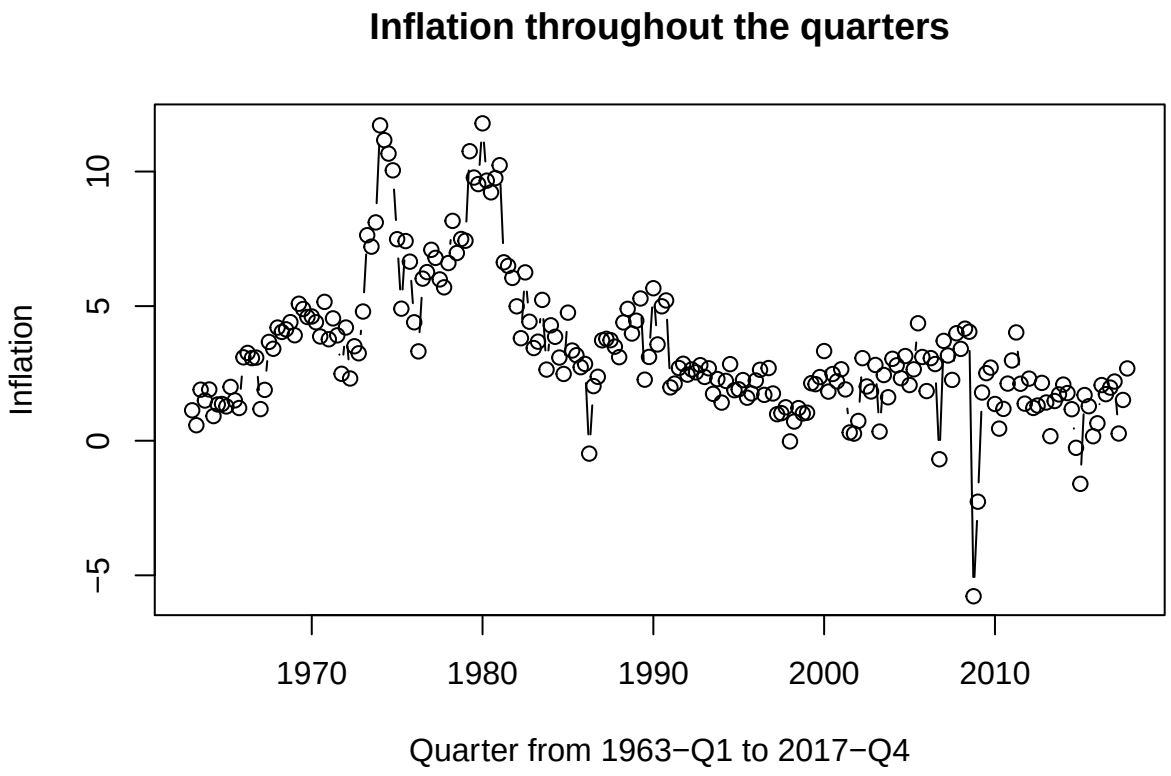
i.

```
data <- data %>% filter(data$freq >= '1962-07-01')
data$Inflat <- c(NA,400*diff(log(data$PCED)))
```

The unit of the Inflation we just computed is in percentage per year as it represents the growth rate in the price index for personnal consumption expenditures. As we are multiplying by the log differences with 400 instead of 100 we are annualizing quarterly growth rates.

ii.

```
data1 <- data %>% filter(data$freq >= "1963-01-01")
plot(
  data1$freq,data1$Inflat, type="b",
  xlab = "Quarter from 1963-Q1 to 2017-Q4" ,
  ylab = "Inflation",
  main = "Inflation throughout the quarters")
```



Based on the plot an upward trend through 1981/1982 is visible. This was followed by a downward trend thereafter. Overall the trend seem deterministic, hence not stochastic. From an historic perspective, this is when tighter monetary policy set by Paul Volcker helped decrease inflation rates.

(b)

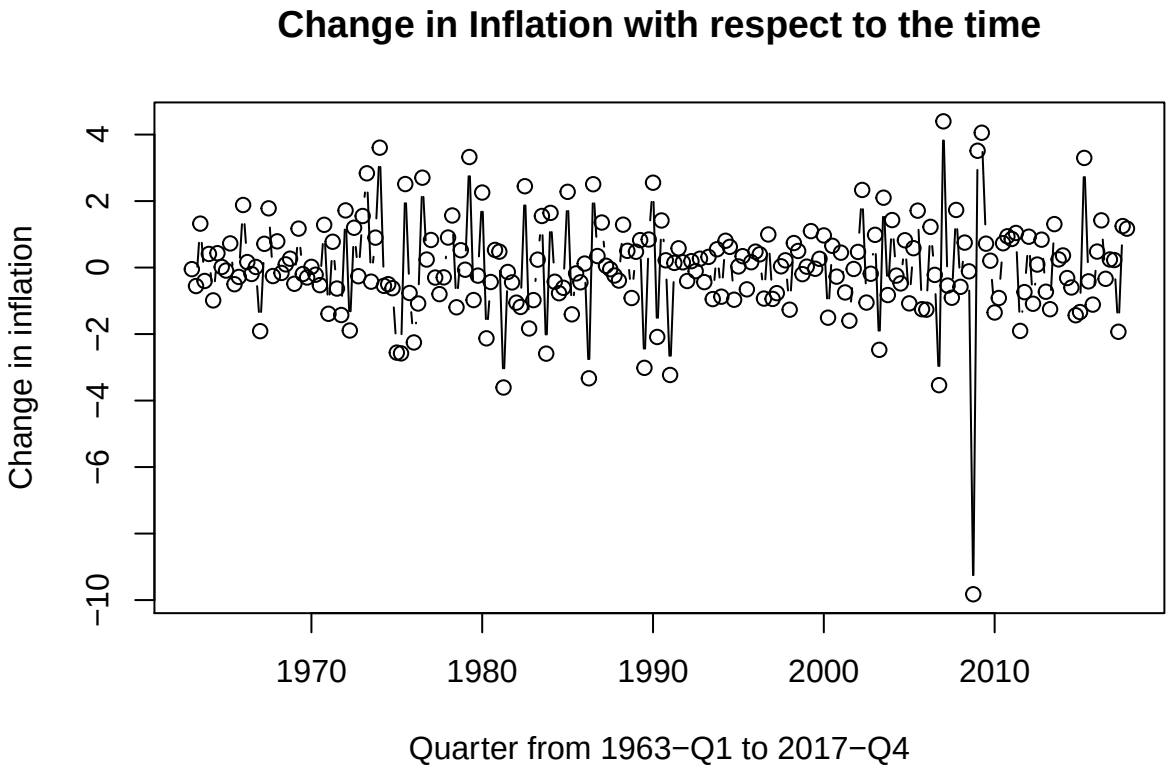
i

```
data$DeltaInflat <- c(NA,diff(data$Inflat))
data$DeltaInflat[is.na(data$DeltaInflat)] <- 0
data2 <- data %>% filter(data$freq >= "1963-01-01")
acf(data$DeltaInflat, lag.max = 4, plot=FALSE)
```

```
##
## Autocorrelations of series 'data$DeltaInflat', by lag
##
##      0      1      2      3      4
##  1.000 -0.246 -0.204  0.136 -0.085
```

ii

```
plot( data2$freq ,data2$DeltaInflat, type="b",
      xlab = "Quarter from 1963-Q1 to 2017-Q4",
      ylab = "Change in inflation",
      main="Change in Inflation with respect to the time")
```



The first order autocorrelation computed in i. shows a negative correlation in the change in inflation from time t-1 to time t. This is in line with the plot above as we can see that consecutive inflation changes have different signs.

(c)

i.

```
INFAR1 <- dynlm(ts(data2$DeltaInflat)~L(ts(data2$DeltaInflat,1)))
summary(INFAR1)
```

```
##
## Time series regression with "ts" data:
## Start = 2, End = 220
##
## Call:
## dynlm(formula = ts(data2$DeltaInflat) ~ L(ts(data2$DeltaInflat,
##     1)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8608 -0.6949  0.0565  0.7307  4.9139
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.007476   0.097416   0.077 0.938897
## L(ts(data2$DeltaInflat, 1)) -0.246680   0.065886  -3.744 0.000232 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.442 on 217 degrees of freedom
## Multiple R-squared:  0.06068,    Adjusted R-squared:  0.05635
## F-statistic: 14.02 on 1 and 217 DF,  p-value: 0.000232
```

The absolute t-statistic of the change in inflation at time t-1 is (|-3.744|) larger than the critical value at 1% significance (~|2.576|). Therefore we reject the null hypothesis at 99% significance and can say that the change in inflation at time t can be explained by the change of inflation at time t-1 with an estimated beta of -0.24668 and an intercept of 0.007476. This beta resembles the first order autocorrelation. R-squared is at 0.06068.

ii.

```
INFAR2 <- dynlm(ts(data2$DeltaInflat) ~ L(ts(data2$DeltaInflat,1))
               + L(ts(data2$DeltaInflat,2)))
summary(INFAR2)
```

```
##
## Time series regression with "ts" data:
## Start = 3, End = 220
##
## Call:
## dynlm(formula = ts(data2$DeltaInflat) ~ L(ts(data2$DeltaInflat,
##     1)) + L(ts(data2$DeltaInflat, 2)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6570 -0.6634  0.0442  0.6909  3.7623
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.009049   0.094039   0.096    0.923
## L(ts(data2$DeltaInflat, 1)) -0.317748   0.065540  -4.848 2.39e-06 ***
## L(ts(data2$DeltaInflat, 2)) -0.284351   0.065647  -4.332 2.27e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.388 on 215 degrees of freedom
## Multiple R-squared:  0.1361, Adjusted R-squared:  0.1281
## F-statistic: 16.94 on 2 and 215 DF,  p-value: 1.472e-07
```

The absolute t-statistic of the change in inflation at time t-1 (|-4.848|) and t-2 (|-4.332|) are larger than the critical value at 1% significance (~|2.576|). Therefore we reject the null hypothesis at 99% significance and can say that the change in inflation at time t can be explained by the change of inflation at time t-1 and t-2 with estimated betas of -0.317748 and -0.284351 respectively and an intercept of 0.009049. This beta resembles the first order autocorrelation. R-squared is at 0.1361, which is significantly higher than in i. (0.06068). Adjusted R-squared also rose from 0.05635 to 0.1281. Including 2 lags reflects the consecutive fluctuative nature of the change in inflation rate, which is showcased by the plot above and the autocorrelations.

iii.

```
#First compute the AR processes with lags n = 1 to 8
INFAR3 <- dynlm(ts(data2$DeltaInflat) ~ L(ts(data2$DeltaInflat,1))+
```

```
                  L(ts(data2$DeltaInflat,2))+L(ts(data2$DeltaInflat,3)))

INFAR4 <- dynlm(ts(data2$DeltaInflat) ~ L(ts(data2$DeltaInflat,1))+
                  L(ts(data2$DeltaInflat,2))+L(ts(data2$DeltaInflat,3))
                +L(ts(data2$DeltaInflat,4)))

INFAR5 <- dynlm(ts(data2$DeltaInflat) ~ L(ts(data2$DeltaInflat,1))+
                  L(ts(data2$DeltaInflat,2))+L(ts(data2$DeltaInflat,3))
                +L(ts(data2$DeltaInflat,4))+L(ts(data2$DeltaInflat,5)))

INFAR6 <- dynlm(ts(data2$DeltaInflat) ~ L(ts(data2$DeltaInflat,1))+
                  L(ts(data2$DeltaInflat,2))+L(ts(data2$DeltaInflat,3))
                +L(ts(data2$DeltaInflat,4))+L(ts(data2$DeltaInflat,5))
                +L(ts(data2$DeltaInflat,6)))

INFAR7 <- dynlm(ts(data2$DeltaInflat) ~ L(ts(data2$DeltaInflat,1))+
                  L(ts(data2$DeltaInflat,2))+L(ts(data2$DeltaInflat,3))
                +L(ts(data2$DeltaInflat,4))+L(ts(data2$DeltaInflat,5))
                +L(ts(data2$DeltaInflat,6))+L(ts(data2$DeltaInflat,7)))

INFAR8 <- dynlm(ts(data2$DeltaInflat) ~ L(ts(data2$DeltaInflat,1))+
                  L(ts(data2$DeltaInflat,2))+L(ts(data2$DeltaInflat,3))
                +L(ts(data2$DeltaInflat,4))+L(ts(data2$DeltaInflat,5))
                +L(ts(data2$DeltaInflat,6))+L(ts(data2$DeltaInflat,7))
                +L(ts(data2$DeltaInflat,8)))

BIC(INFAR1,INFAR2,INFAR3,INFAR4,INFAR5,INFAR6,INFAR7,INFAR8)
```

```
##        df      BIC
## INFAR1  3 795.8622
## INFAR2  4 780.2706
## INFAR3  5 782.4807
## INFAR4  6 782.5262
## INFAR5  7 782.0835
## INFAR6  8 782.6035
## INFAR7  9 785.3662
## INFAR8 10 787.8029
```

BIC goes for the second lag as it is the smallest.

```
AIC(INFAR1,INFAR2,INFAR3,INFAR4,INFAR5,INFAR6,INFAR7,INFAR8)
```

```
##        df      AIC
## INFAR1  3 785.6950
## INFAR2  4 766.7326
## INFAR3  5 765.5812
## INFAR4  6 762.2745
## INFAR5  7 758.4890
## INFAR6  8 755.6757
## INFAR7  9 755.1145
## INFAR8 10 754.2371
```

AIC goes for the lag 8, it makes sense as it is supposed to take larger lags.

iv.

```
#Create forecast with AR(2)
#Coefficients:
#                            Estimate Std. Error t value Pr(>|t|)
#(Intercept)                 0.009049   0.094039   0.096    0.923
#L(ts(data2$DeltaInflat, 1)) -0.317748   0.065540  -4.848 2.39e-06 ***
#L(ts(data2$DeltaInflat, 2)) -0.284351   0.065647  -4.332 2.27e-05 ***

#Value of deltainf2017Q4 is :

Q4 <- data2$DeltaInflat[220]
Q3 <- data2$DeltaInflat[219]
#Q3 <- data2$DeltaInflat[data2$freq == "2017-07-01"]

Deltainf2018Q1 <- 0.009049 + -0.317748* Q4 -0.284351 * Q3
Deltainf2018Q1
```

```
## [1] -0.7174659
```

```
#Negative change expected with lag 2
```

v.

```
#Value of inflation Q4 2017

InfQ42017 <- data2$Inflat[220]
InfQ12018 <- InfQ42017 + Deltainf2018Q1
InfQ12018
```

```
## [1] 1.967405
```

The forecasted value of the inflation is equal to the previous level of inflation plus the predicted change in inflation between Q4 2017 and Q1 2018.

(d)

i.

```
#Use ADF test for the regression
#First create the function

DeltaInf_t2 <- dynlm(ts(data2$DeltaInflat)~L(ts(data2$DeltaInflat,1))
                     +L(ts(data2$DeltaInflat,2))+L(ts(data2$Inflat,1)))

coeftest(DeltaInf_t2, vcov = vcov)
```

```
##
## t test of coefficients:
##
##                              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                  0.370881   0.160122  2.3162 0.0214913 *
## L(ts(data2$DeltaInflat, 1)) -0.253185   0.068625 -3.6894 0.0002851 ***
```

```
## L(ts(data2$DeltaInflat, 2)) -0.240765   0.066538 -3.6185 0.0003698 ***
## L(ts(data2$Inflat, 1))      -0.107707   0.038882 -2.7701 0.0060966 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#ADF test unit root for Infl(t-1) is
ADFinf1 <- -0.107707/0.038882
ADFinf1
```

```
## [1] -2.770099
```

The Augmented Dickey-Fuller test for the null hypothesis of a unit root of a univarate time series. Here the ADF t-test is which help us reject the null hypothesis at a 10% level of significance (- 2.86 <-2.770099<-2.57) and without time trend. We used homoskedasticity-only standard errors as proposed in page 586 in Stock and Watson.

ii.

```
#create a second function with the time trend
DeltaInf_t_timetrend <- dynlm(ts(data2$DeltaInflat)~ts(trend(data2$DeltaInflat,scale=FALSE))
                              +L(ts(data2$DeltaInflat,1))+L(ts(data2$DeltaInflat,2))
                              +L(ts(data2$Inflat,1)))

coeftest(DeltaInf_t_timetrend, vcov = vcov)
```

```
##
## t test of coefficients:
##
##                                            Estimate Std. Error t value
## (Intercept)                                0.9531242  0.3059669  3.1151
## ts(trend(data2$DeltaInflat, scale = FALSE)) -0.0037429  0.0016818 -2.2256
## L(ts(data2$DeltaInflat, 1))                -0.2286588  0.0688867 -3.3193
## L(ts(data2$DeltaInflat, 2))                -0.2266336  0.0662369 -3.4216
## L(ts(data2$Inflat, 1))                     -0.1568004  0.0443957 -3.5319
##                                             Pr(>|t|)
## (Intercept)                                0.0020917 **
## ts(trend(data2$DeltaInflat, scale = FALSE)) 0.0270904 *
## L(ts(data2$DeltaInflat, 1))                0.0010613 **
## L(ts(data2$DeltaInflat, 2))                0.0007464 ***
## L(ts(data2$Inflat, 1))                     0.0005058 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ADFinf2 <- -0.1568004/0.0443957
ADFinf2
```

```
## [1] -3.531883
```

With the time trend and intercept, the ADF t-test is -3.531883. This mean that we can reject the null hypothesis at a 5% level in favor (-3.41 > -3.531883 > -3.96) for the alternative hypothesis, which is that Delta_Inf_t is stationary around a deterministic linear time trend. The addition of a deterministic time trend decreaases regresssion intercept, but increases standard error. However, as we can reject H0 at a lower significance level, we would prefer the second equation.

iii.

```
BIC(INFAR1,INFAR2,INFAR3,INFAR4,INFAR5,INFAR6,INFAR7,INFAR8)
```

```
##        df      BIC
## INFAR1  3 795.8622
## INFAR2  4 780.2706
## INFAR3  5 782.4807
## INFAR4  6 782.5262
## INFAR5  7 782.0835
## INFAR6  8 782.6035
## INFAR7  9 785.3662
## INFAR8 10 787.8029
```

BIC is minimized for lag = 2. According to this criterion, we should not add or remove any lags.

```
AIC(INFAR1,INFAR2,INFAR3,INFAR4,INFAR5,INFAR6,INFAR7,INFAR8)
```

```
##        df      AIC
## INFAR1  3 785.6950
## INFAR2  4 766.7326
## INFAR3  5 765.5812
## INFAR4  6 762.2745
## INFAR5  7 758.4890
## INFAR6  8 755.6757
## INFAR7  9 755.1145
## INFAR8 10 754.2371
```

However,according to Stock and Watson (page 588/589), studies of the ADF statistic suggest more lages are better than too few. Therefore they recommend to use the AIC instead of the BIC to estimate the number of lags for the ADF statistic. For AIC, increasing lag size is preferrable. We tried until 8 lags. Optimal lag size might be even more. One issue might have been that we did not include all periods for all lags. As AIC tends to overestimate lag size with nonzero probability for large sample sizes, the best pick probably lies somewhere inbetween. Fewer lags definitely not, more lags for sure.

iv.

ADF H0 rejection does not necessarily mean that the time series has a root. It rather indicattes that there is insufficient evidence to reject the null hypothesis. This is shown by the fact that adding a deterministic trend adds significance to rejecting H0.

(e)

```
#QLR test

# set up a range of possible break dates

qlr_data <- data2[floor(220*0.15):( nrow(data2) - floor(220*0.15) ),]
tau <- qlr_data$freq
# initialize vector of F-statistics
Fstats <- numeric(length(tau))

DeltaINF_ts<- ts(data$DeltaInflat)
```

```
# estimation loop over break dates
for(i in 1:length(tau)) {

  # set up dummy variable
         D <- ts((data2$freq > tau[i]))

    test <- dynlm(DeltaINF_ts ~
                    L(DeltaINF_ts,1) +
                    L(DeltaINF_ts, 2)+
                    D*L(DeltaINF_ts,1) +
                    D*L(DeltaINF_ts, 2))

    Fstats[i] <- linearHypothesis(test,
                                  c("DTRUE"),
                                  vcov. = sandwich)$F[2]

}
max(Fstats)
```

```
## [1] 3.616541
```

```
as.yearqtr(tau[which.max(Fstats)])
```

```
## [1] "1974 Q4"
```

The QLR test with 15% trimming to test the stability of the coefficients in the AR(2) model for delta inflation has its highest F-statistic at Q4 1974 of 3.616541. Based on this test the AR(2) seems stable as the 10% critical value for 3 restrictions is 4.09. However, looking at the plot we think we made an error in the linearHypothesis() function. Based on the plot a break is suspected in 1971 or 1982.

Reference: Stock and Watson, Ch. 15.

Question 2

```
us_macro_monthly1 <- read_excel("data/us_macro_monthly1.xlsx")

#data_set <- read_excel("data/us_macro_quarterly1.xlsx")
us_macro_monthly<-subset(us_macro_monthly1, freq> "1959-01-01" & freq < "2004-12-02")
tail(us_macro_monthly)
```

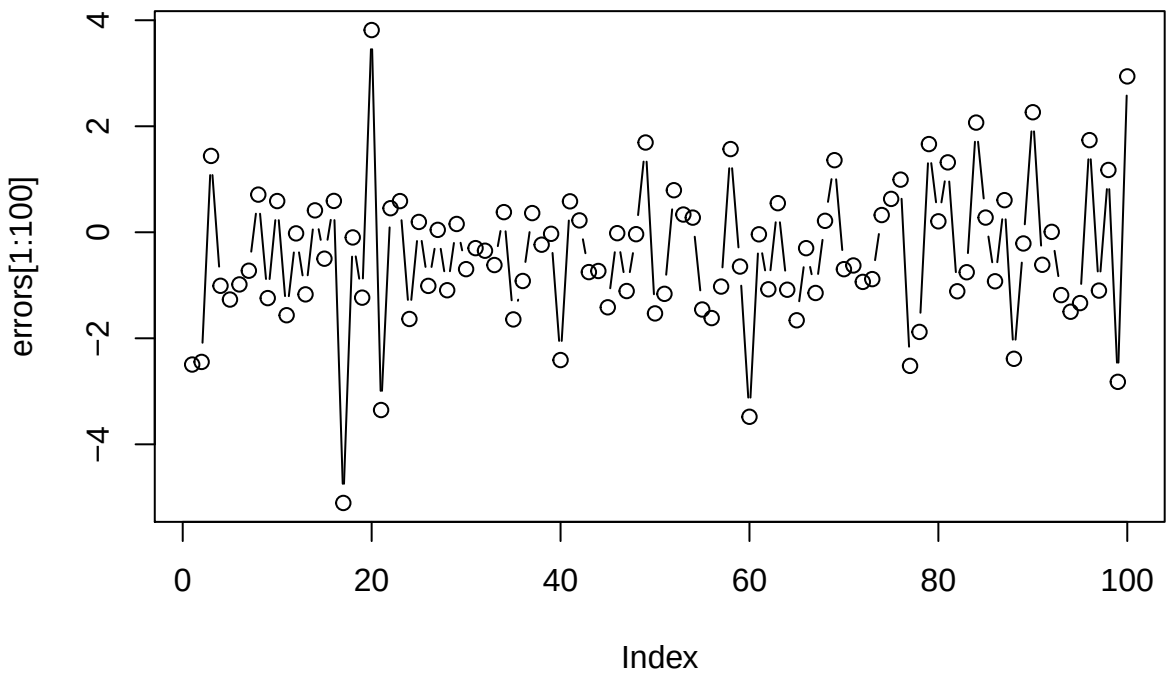
```
## # A tibble: 6 x 13
##   freq                 CPI EXUSUK FEDFUNDS   GS1  GS10 INDPRO  PCED TB3MS
##   <dttm>             <dbl>  <dbl>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1 2004-07-01 00:00:00  189.   1.84     1.26  2.1   4.5    96.6  89.8  1.33
## 2 2004-08-01 00:00:00  189.   1.82     1.43  2.02  4.28   96.7  89.9  1.48
## 3 2004-09-01 00:00:00  190.   1.79     1.61  2.12  4.13   96.7  90.1  1.65
## 4 2004-10-01 00:00:00  191.   1.81     1.76  2.23  4.1    97.7  90.4  1.76
## 5 2004-11-01 00:00:00  192.   1.86     1.93  2.5   4.19   97.8  90.7  2.07
## 6 2004-12-01 00:00:00  192.   1.93     2.16  2.67  4.23   98.5  90.8  2.19
## # ... with 4 more variables: UNRATE <dbl>, WPU0561 <dbl>, PAYEMS <dbl>,
## #   DJIA <dbl>
```

Importing the data.

```
us_macro_monthly$pi_CPI <- 1200*log(us_macro_monthly$CPI/lag(us_macro_monthly$CPI, 1))
us_macro_monthly$pi_PCED <- 1200*log(us_macro_monthly$PCED/lag(us_macro_monthly$PCED, 1))
us_macro_monthly$Y_t <- us_macro_monthly$pi_CPI -us_macro_monthly$pi_PCED
```

Calculating monthly rates of price inflation using CPI and PCED. Then, finding our Y.

```
y_t <- (na.omit(us_macro_monthly$Y_t))
errors <- y_t-mean(y_t)
plot(errors[1:100],type='b')
```



We believe that errors will be serially correlated, that is, they will follow some sort of pattern rather than being random. By plotting standard errors we can see that there is probably negative autocorrelation between error terms as positive errors are usually followed by negative ones and vice versa. Therefore we expect the OLS estimator to be unbiased and consistent, but inefficient. If we want to test for serial correlation of error terms more precisely we can conduct a Durbin-Watson or Ljung Box tests.

##(a)

```
k <- function(z){
  if(z <= 1 && z >= 0){
    return(1-z)
  }
  else{
    return(0)
  }
}
```

Now we define function k, which is the Bertlett kernel described in the question.

```
t<-length(y_t)
sample_auto_cov <- acf(y_t,lag.max =1000,type="covariance", plot=F)

#b <- t-1
b <- 0.75*t^(1/3)
```

```
h <- seq(1, t-1, by=1)

w_hat_squared <- sample_auto_cov$acf[1] +  2* sum(sapply(h/b, k) * sample_auto_cov$acf[-1] )

t_statistic_a <- ((t^(0.5)*(mean(y_t))-0)/((w_hat_squared)^(0.5)))
```

In this step we are calculating our Newey-West HAC standard errors.

```
#mean(y_t)
#w_hat_squared**(0.5)
#t_statistic_a

CI <- c(mean(y_t) -1.96* (w_hat_squared**(0.5)/t^(0.5)), mean(y_t)
        +1.96* (w_hat_squared**(0.5)/t**(0.5)))
CI
```

```
## [1] 0.3167889 0.6491101
```

We can see that 95% confidence interval for mu is CI=(0.3130207, 0.6460639).

In the Newey-West HAC standard errors approach, bandwidth paramater represents the number of auto-correlations to include in the estimate. In order to get asymptotic normality consistency we need to make sure that our bandwidth grows slower than T. Rule for choice for bandwidth parameter derived by Donald Andrews is some fraction of [T**(1/3)]. Therefore we will use 0.75T^(1/3).

There is a statistical evidence that the mean inflation rate for CPI is grater than the rate for the PCED. This comes from the fact that the whole CI=(0.3130207, 0.6460639) lies above 0. Therefore, there is very high probability that CI lies above 0, and hence that CPI rate is higher than PCED rate.

(b)

As mentioned in the previous part, if we choose our bandwidth paramater such that it grows slower than T, that is b(T)/T -> 0, we will have that our standard errors are consistent for unknown standard deviations of our OLS estimators.This parameter is our choice therefore we can control that we satisfy this condition.

(c)

```
to_chunk <-function(x,n) split(x, cut(seq_along(x), n, labels = FALSE))
```

```
q <-  4
chunks = to_chunk(y_t,q)
chunks_df <- do.call(rbind.data.frame, chunks)
chunks_df$means <- rowMeans(chunks_df)
var_chunk<- var(chunks_df$means)
sqrt(q)*mean(chunks_df$means)/sqrt(var_chunk)
```

```
## [1] 3.686488
```

```
t_statistic_a
```

```
## [1] 5.696785
```

We get 3.686488 for our t-statistic and therefore we are able to reject the null at 5%, therefore obtaining that mu>0.In the result from (a) we got to the same conclusion.

(d)

Our goal was to provide an approach that works under wide dependence types so that we do no need to assume some particular dependence structure. Simulation shows that choice of q=4 or q=8 performs the best.

When data is split into q groups, we need 2 assumptions in order to be able to use our approach: 1. Group estimates are asymptotically independent 2. Group estimates are asymptotically normal.

We believe that both assumptions are satisfied, therefore we can use t statistic approach to test our hypothesis.

Reference: Stock and Watson, Ch. 16. Hamilton (1994), Ch. 10 Newey, W. and West, K. (1987). A simple positive semi-definite, heteroskedastic and autocorrelation consistent covariance matrix. Econometrica 55, 703-708. Andrews, D. W. K. (1991). Heteroskedasticity and autocorrelation consistent covariance matrix estimation. Econometrica 59, 817-858.

Question 3

(a)

We used the daily return of S&P500 from 1 January 2000 to 15 November 2019 as our dataset (5000 observations). We have computed the log-log rank-size regression and Hill's estimates at truncation levels of 10%, 5% and 1%. The result is as follows:

```
options("getSymbols.warning4.0"=FALSE)
options("getSymbols.yahoo.warning"=FALSE)

# Downloading S&P500 price using quantmod (create 5000 observations)
getSymbols("^GSPC", from = '2000-01-01',
           to = "2019-11-15",warnings = FALSE,
           auto.assign = TRUE)
```

```
## [1] "^GSPC"
```

```
tail(GSPC)
```

```
##            GSPC.Open GSPC.High GSPC.Low GSPC.Close GSPC.Volume GSPC.Adjusted
## 2019-11-07   3087.02   3097.77  3080.23    3085.18  4144640000       3085.18
## 2019-11-08   3081.25   3093.09  3073.58    3093.08  3499150000       3093.08
## 2019-11-11   3080.33   3088.33  3075.82    3087.01  3035530000       3087.01
## 2019-11-12   3089.28   3102.61  3084.73    3091.84  3466010000       3091.84
## 2019-11-13   3084.18   3098.06  3078.80    3094.04  3509280000       3094.04
## 2019-11-14   3090.75   3098.20  3083.26    3096.63  3276070000       3096.63
```

```
SP500 <- GSPC[,c(1,4)]

# Computer Daily Return
SP500 <- SP500[,c(1,2)]
SP500.df = data.frame(Date = index(SP500), coredata(SP500))

SP500.df <- SP500.df %>% mutate(daily_return = (GSPC.Close-GSPC.Open)/GSPC.Open)

## Log-Log Rank-Size Regression
data_summary <- SP500.df %>% mutate(rank = rank(-daily_return))
```

```r
data_summary <- data_summary %>% arrange(-daily_return)

# 10% truncation (n = 500)
data_summary_500 <- data_summary[1:500,]

lm_loglog_500 <- lm(log(rank) ~ log(daily_return), data = data_summary_500)
summary(lm_loglog_500)
```

```
##
## Call:
## lm(formula = log(rank) ~ log(daily_return), data = data_summary_500)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.94232 -0.05980  0.01264  0.08653  0.23215
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -4.46817    0.06187  -72.22   <2e-16 ***
## log(daily_return) -2.42360    0.01540 -157.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1371 on 498 degrees of freedom
## Multiple R-squared:  0.9803, Adjusted R-squared:  0.9803
## F-statistic: 2.478e+04 on 1 and 498 DF,  p-value: < 2.2e-16
```

```r
# 5% truncation (n = 250)
data_summary_250 <- data_summary[1:250,]
lm_loglog_250 <- lm(log(rank) ~ log(daily_return), data = data_summary_250)
summary(lm_loglog_250)
```

```
##
## Call:
## lm(formula = log(rank) ~ log(daily_return), data = data_summary_250)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68184 -0.08851  0.01743  0.07167  0.25698
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -5.55597    0.08764   -63.4   <2e-16 ***
## log(daily_return) -2.73164    0.02362  -115.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1299 on 248 degrees of freedom
## Multiple R-squared:  0.9818, Adjusted R-squared:  0.9817
## F-statistic: 1.338e+04 on 1 and 248 DF,  p-value: < 2.2e-16
```

```r
# 1% truncation (n = 50)
data_summary_50 <- data_summary[1:50,]
lm_loglog_50 <- lm(log(rank) ~ log(daily_return), data = data_summary_50)
```

```
## Hill's estimate

# 10% truncation (n = 500)
data_summary_500 <- data_summary_500 %>% mutate(diff = log(daily_return)-log(last(daily_return)))
hill_500 <- 500/sum(data_summary_500$diff)

# 5% truncation (n = 250)
data_summary_250 <- data_summary_250 %>% mutate(diff = log(daily_return)-log(last(daily_return)))
hill_250 <- 250/sum(data_summary_250$diff)

# 1% truncation (n = 50)
data_summary_50 <- data_summary_50 %>% mutate(diff = log(daily_return)-log(last(daily_return)))
hill_50 <- 50/sum(data_summary_50$diff)
```

We used the daily return of S&P500 from 1 January 2000 to 15 November 2019 as our dataset (5000 observations). We have computed the log-log rank-size regression (left) and Hill's estimates (right) at truncation levels of 10%, 5% and 1%. The result is as follows:

Trunc	Estimates	Trunc	Estimates
10%	2.423602525	10%	2.088717785
5%	2.731644101	5%	2.407522652
1%	3.410261255	1%	3.714418265

(b)

```
## Log-Log Rank-Size Regression Confidence Interval
t_99 <- qt(p=.005, df=Inf, lower.tail=FALSE)
t_95 <- qt(p=.025, df=Inf, lower.tail=FALSE)
t_90 <- qt(p=.05, df=Inf, lower.tail=FALSE)

# 10% truncation (n = 500)
sum1 <- summary(lm_loglog_500)
est_loglog_500 <- -sum1$coefficients[2,1]
se_loglog_500 <- sum1$coefficients[2,2]

CI99_loglog_500 <- c(est_loglog_500-t_99*se_loglog_500*sqrt(2),est_loglog_500
                     +t_99*se_loglog_500*sqrt(2))
CI95_loglog_500 <- c(est_loglog_500-t_95*se_loglog_500*sqrt(2),est_loglog_500
                     +t_95*se_loglog_500*sqrt(2))
CI90_loglog_500 <- c(est_loglog_500-t_90*se_loglog_500*sqrt(2),est_loglog_500
                     +t_90*se_loglog_500*sqrt(2))

# 5% truncation (n = 250)
sum2 <- summary(lm_loglog_250)
est_loglog_250 <- -sum2$coefficients[2,1]
se_loglog_250 <- sum2$coefficients[2,2]

CI99_loglog_250 <- c(est_loglog_250-t_99*se_loglog_250*sqrt(2),
                     est_loglog_250+t_99*se_loglog_250*sqrt(2))
CI95_loglog_250 <- c(est_loglog_250-t_95*se_loglog_250*sqrt(2),
                     est_loglog_250+t_95*se_loglog_250*sqrt(2))
```

```
CI90_loglog_250 <- c(est_loglog_250-t_90*se_loglog_250*sqrt(2),
                     est_loglog_250+t_90*se_loglog_250*sqrt(2))

# 1% truncation (n = 50)
sum3 <- summary(lm_loglog_50)
est_loglog_50 <- -sum3$coefficients[2,1]
se_loglog_50 <- sum3$coefficients[2,2]

CI99_loglog_50 <- c(est_loglog_50-t_99*se_loglog_50
                    *sqrt(2),est_loglog_50+t_99*se_loglog_50*sqrt(2))
CI95_loglog_50 <- c(est_loglog_50-t_95*se_loglog_50
                    *sqrt(2),est_loglog_50+t_95*se_loglog_50*sqrt(2))
CI90_loglog_50 <- c(est_loglog_50-t_90*se_loglog_50*
                     sqrt(2),est_loglog_50+t_90*se_loglog_50*sqrt(2))

## Hill's estimate Confidence Interval

# 10% truncation (n = 500)
se_hill_500 <- hill_500/sqrt(500)

CI99_hill_500 <- c(hill_500-t_99*se_hill_500,hill_500+t_99*se_hill_500)
CI95_hill_500 <- c(hill_500-t_95*se_hill_500,hill_500+t_95*se_hill_500)
CI90_hill_500 <- c(hill_500-t_90*se_hill_500,hill_500+t_90*se_hill_500)

# 5% truncation (n = 250)
se_hill_250 <- hill_250/sqrt(250)

CI99_hill_250 <- c(hill_250-t_99*se_hill_250,hill_250+t_99*se_hill_250)
CI95_hill_250 <- c(hill_250-t_95*se_hill_250,hill_250+t_95*se_hill_250)
CI90_hill_250 <- c(hill_250-t_90*se_hill_250,hill_250+t_90*se_hill_250)

# 1% truncation (n = 50)
se_hill_50 <- hill_50/sqrt(50)

CI99_hill_50 <- c(hill_50-t_99*se_hill_50,hill_50+t_99*se_hill_50)
CI95_hill_50 <- c(hill_50-t_95*se_hill_50,hill_50+t_95*se_hill_50)
CI90_hill_50 <- c(hill_50-t_90*se_hill_50,hill_50+t_90*se_hill_50)
```

The confidence interval at constructed at 99%, 95% and 90%. The result is as follows:

Log-log rank-size regression:

CI/Trunc	10%	5%	1%
99%	(2.368,2.480)	(2.646,2.818)	(3.164,3.657)
95%	(2.381,2.466)	(2.666,2.797)	(3.223,3.598)
90%	(2.388,2.459)	(2.677,2.787)	(3.253,3.568)

Hill's estimates:

CI/Trunc	10%	5%	1%
99%	(1.848,2.329)	(2.015,2.800)	(2.361,5.067)
95%	(1.906,2.272)	(2.109,2.706)	(2.685,4.744)

CI/Trunc	10%	5%	1%
90%	(1.935,2.242)	(2.157,2.658)	(2.850,4.578)

(c)

In lecture, we learnt that the tail index is $2<\zeta<4$ for developed markets. As we can see from above, all estimates of lies in the interval. Regarding the confidence intervals, all of them intersects with the range of . We can also see, as the truncation level reduces (i.e. moving to a smaller n), the estimates of tail index increases and this implies a lighter tail.

Regarding the finiteness of variances for time series, ζ has to be greater than 2 for finite variances. All of the data are showing $\zeta>2$ except for the 10% truncation of Hill's estimate, the lower bound of its confidence intervals are slightly below 2. However, these figures are still very close to 2 and therefore most of the evidence suggests finite variances.