Assignment 05, Question 1

Jeff Nguyen

24/04/2021

University of Southern California
Marshall School of Business
FBE 543 Forecasting and Risk Analysis

#**Question 1**

```
library(quantmod)

# Set start date and end date of data
start_date <- "2014-01-01"
end_date <- "2021-03-24"

# Get data
getSymbols("JPM", src = "yahoo", , from = start_date, to = end_date)
```

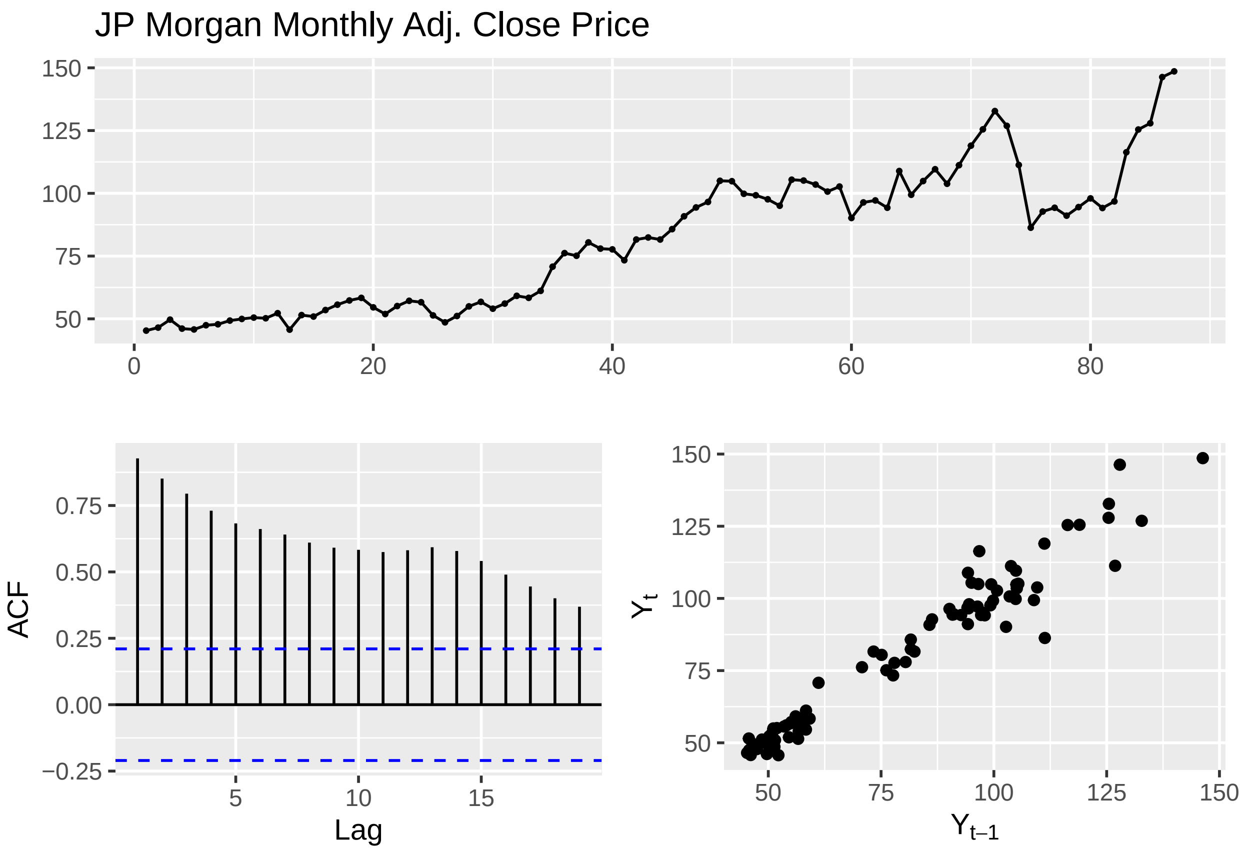
```
## [1] "JPM"
```

```
adjJPM_mo <- to.monthly(JPM)$JPM.Adjusted  # Monthly Adjusted Closing Price
rJPM_mo <- diff(log(adjJPM_mo))[-1]  # Monthly Returns
```

Data Observation:

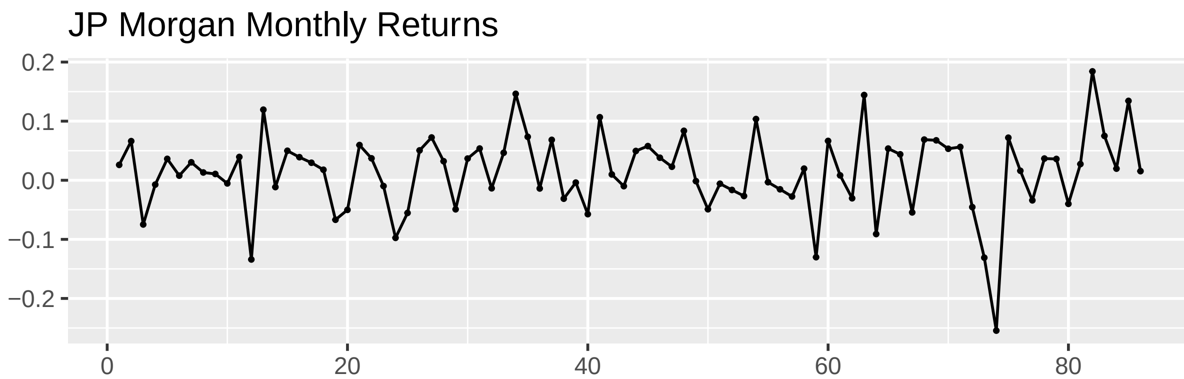
Observing monthly adj. closing prices:

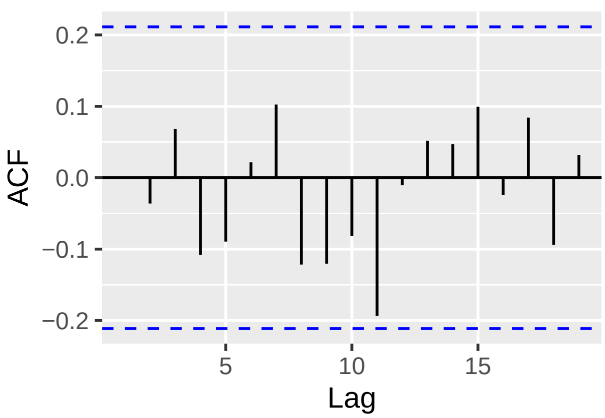
```
library(forecast)

ggtsdisplay(adjJPM_mo, main="JP Morgan Monthly Adj. Close Price", plot.type="scatter")
```

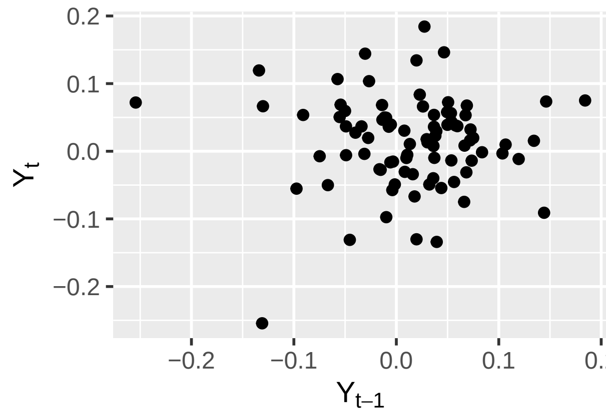


Observing monthly returns:

```
ggtsdisplay(rJPM_mo, main="JP Morgan Monthly Returns", plot.type="scatter")
```







Remarks
We can see JPM's monthly adjusted closing price' lag plots exhibit a linear pattern, implying that the data is strongly non-random and thus, a first-order autoregressive model might be appropriate.

$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t \tag{1}$$

On the other hand, JPM's monthly returns's lag plot does not exhibit any obvious patterns, implying that the data is strongly random.

1. Test for the stationarity of the adjusted closing prices for JPM.

We run Augmented Dickey Fuller Test for JPM. Recall that the null hypothesis for Dickey-Fuller Test is that a unit root is present in our autoregressive model, meaning the variable is a non-stationary variable.

```
library(aTSA)

adf.test(adjJPM_mo)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag  ADF p.value
## [1,]   0 2.39   0.990
## [2,]   1 2.14   0.990
```

```
## [3,]   2 2.08   0.990
## [4,]   3 1.87   0.983
## Type 2: with drift no trend
##      lag  ADF p.value
## [1,]   0 2.15    0.99
## [2,]   1 2.00    0.99
## [3,]   2 2.07    0.99
## [4,]   3 1.75    0.99
## Type 3: with drift and trend
##      lag  ADF p.value
## [1,]   0 3.23    0.99
## [2,]   1 3.13    0.99
## [3,]   2 3.25    0.99
## [4,]   3 3.03    0.99
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

We can observe $p - value = .99 > .05$. Thus, we fail to reject the null hypothesis. In other words, JPM monthly adjusted closing price has a unit root and therefore, is a non-stationary variable.

2. Test for the stationarity of the returns for JPM.

Similarly, we run (A)DF test for JPM's monthly returns:

```r
adf.test(rJPM_mo)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag   ADF p.value
## [1,]   0  8.81    0.99
## [2,]   1 12.46    0.99
## [3,]   2 14.50    0.99
## [4,]   3 16.67    0.99
## Type 2: with drift no trend
##      lag   ADF p.value
## [1,]   0  9.12    0.99
## [2,]   1 13.08    0.99
## [3,]   2 15.36    0.99
## [4,]   3 17.93    0.99
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,]   0  9.08    0.99
## [2,]   1 13.00    0.99
## [3,]   2 15.27    0.99
## [4,]   3 17.82    0.99
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

We can observe $p - value = .99 > .05$. Thus, we fail to reject the null hypothesis. In other words, JPM monthly returns has a unit root and therefore, is a non-stationary variable.

3. Run the best ARIMA model for JPM returns.

We run auto ARIMA:

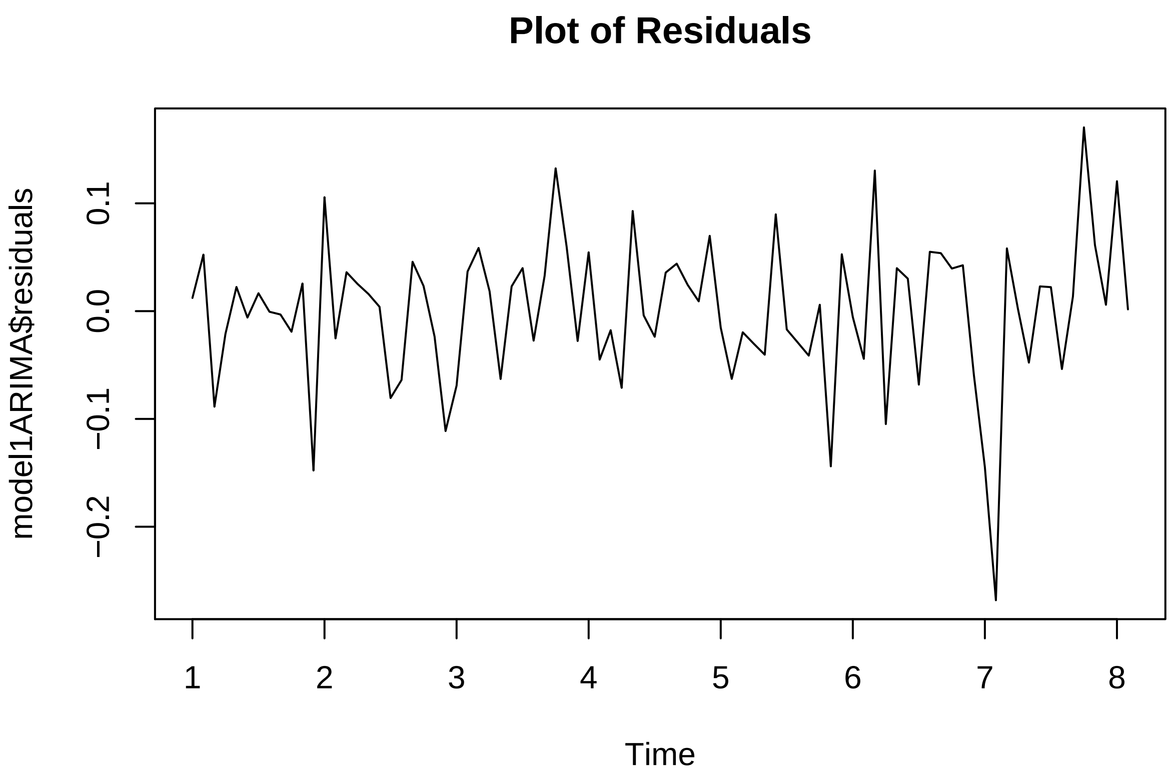
```
model1ARIMA <- auto.arima(rJPM_mo)
summary(model1ARIMA)
```

```
## Series: rJPM_mo
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##         mean
##       0.0138
## s.e.  0.0072
##
## sigma^2 estimated as 0.004526:  log likelihood=110.59
## AIC=-217.18   AICc=-217.03   BIC=-212.27
##
## Training set error measures:
##                        ME       RMSE        MAE      MPE     MAPE      MASE
## Training set 3.987334e-20 0.06688026 0.04967865 117.2023 122.2534 0.6404292
##                      ACF1
## Training set -0.001524886
```

We can see that with $ARIMA(0,0,0)$, we have great $ACF1$ statistics, implying a good fit for forecasting.

4. Test for the existence of heteroskedasticity on the residuals of the JPM's ARIMA model.

```
plot(model1ARIMA$residuals, main="Plot of Residuals")
```



Testing for hetereoskedacity on the residuals of the ARIMA model above:

11. Run an ARCH and/or GARCH model on JPM's returns data.

```
library(rugarch)

# Write Specification of Your GARCH Model using "sGrach" or standard GARCH Mode.
garch1 <- ugarchspec(variance.model=list(model="sGARCH", garchOrder=c(1, 1)), mean.model=list(armaOrder=

# Fit the Model to Data
garch1_rJPM_mo <- ugarchfit(spec=garch1, data=rJPM_mo)
garch1_rJPM_mo
```

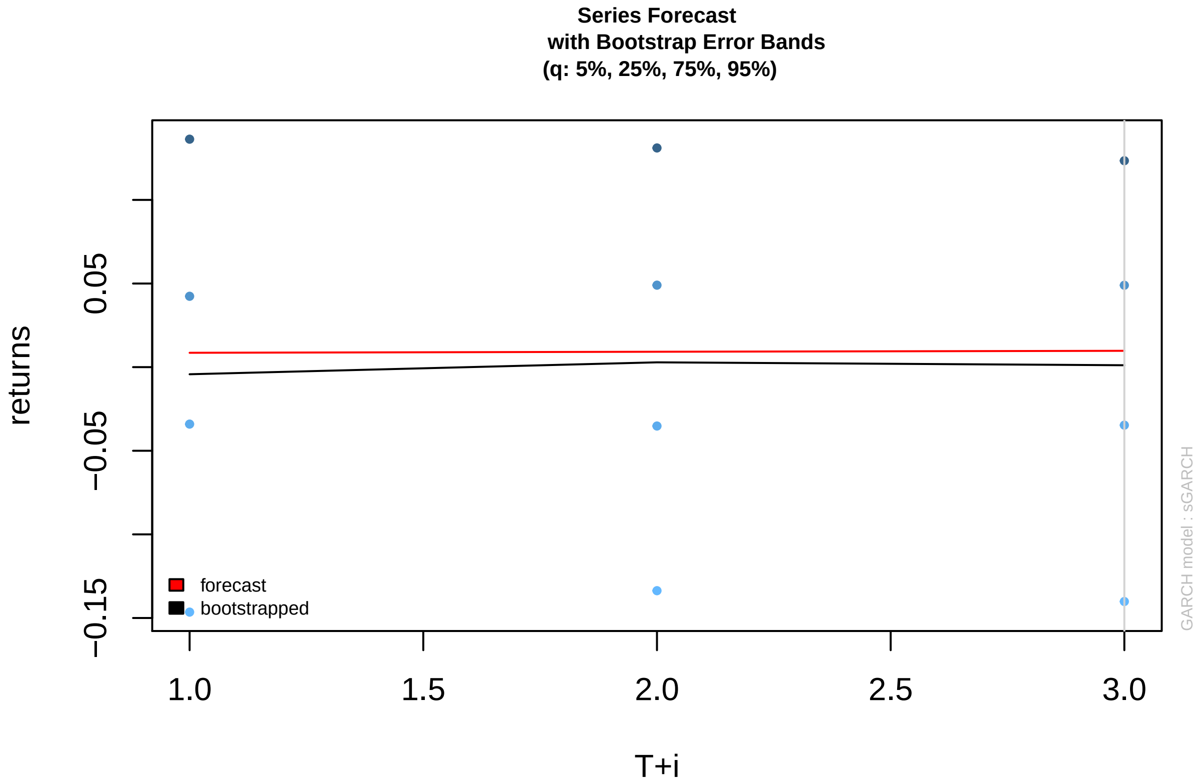
```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(1,0,1)
## Distribution : std
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
```

```
## mu      0.013384    0.002179   6.1417 0.000000
## ar1     0.869056    0.050801  17.1070 0.000000
## ma1    -1.000000    0.068492 -14.6002 0.000000
## omega   0.000094    0.000304   0.3075 0.758463
## alpha1  0.000000    0.028303   0.0000 1.000000
## beta1   0.999000    0.063828  15.6515 0.000000
## shape   2.650918    0.581725   4.5570 0.000005
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.013384    0.002974  4.50106 0.000007
## ar1     0.869056    0.040199 21.61906 0.000000
## ma1    -1.000000    0.111538 -8.96556 0.000000
## omega   0.000094    0.000616  0.15204 0.879152
## alpha1  0.000000    0.126918  0.00000 1.000000
## beta1   0.999000    0.073145 13.65773 0.000000
## shape   2.650918    0.790405  3.35387 0.000797
##
## LogLikelihood : 118.8186
##
## Information Criteria
## ------------------------------------
##
## Akaike       -2.6004
## Bayes        -2.4007
## Shibata      -2.6124
## Hannan-Quinn -2.5200
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.02121  0.8842
## Lag[2*(p+q)+(p+q)-1][5]   0.36813  1.0000
## Lag[4*(p+q)+(p+q)-1][9]   1.04492  0.9997
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.8106  0.3679
## Lag[2*(p+q)+(p+q)-1][5]    3.0154  0.4042
## Lag[4*(p+q)+(p+q)-1][9]    3.6955  0.6408
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.4116 0.500 2.000  0.5211
## ARCH Lag[5]    0.9673 1.440 1.667  0.7427
## ARCH Lag[7]    1.2564 2.315 1.543  0.8686
##
## Nyblom stability test
## ------------------------------------
```

```
## Joint Statistic:  1.5509
## Individual Statistics:
## mu     0.04699
## ar1    0.14262
## ma1    0.18652
## omega  0.06568
## alpha1 0.06573
## beta1  0.05588
## shape  0.06595
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.69 1.9 2.35
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           0.6787 0.49927
## Negative Sign Bias  1.7223 0.08884   *
## Positive Sign Bias  0.7764 0.43979
## Joint Effect        4.2661 0.23413
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     14.00       0.7837
## 2    30     17.18       0.9593
## 3    40     36.50       0.5844
## 4    50     29.16       0.9891
##
##
## Elapsed time : 0.247052
```

12. Do a three-period ahead forecast of the conditional variance.

```r
# Forecast Model
predict_rJPM_mo <- ugarchboot(garch1_rJPM_mo,n.ahead=3, method=c("Partial", "Full")[1])
plot(predict_rJPM_mo, which=2)
```



Series Forecast
with Bootstrap Error Bands
(q: 5%, 25%, 75%, 95%)
returns
0.05
−0.05
−0.15
1.0
1.5
2.0
2.5
3.0
T+i
forecast
bootstrapped
GARCH model : sGARCH