# Problem set 2

학과 : e-비즈니스학과

학번 : 201921527

이름 : 박성우

---

## 0. Load packages and raw data

In [1]:
```
# Load data-preprocessing pacakages
import pandas as pd
import numpy as np

# Load visualization pacakage
import matplotlib.pyplot as plt

# Load modeling pacakages
import pmdarima as pm
import statsmodels.api as sm
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.stattools import kpss
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.tsa.stattools import pacf
from statsmodels.stats.diagnostic import acorr_ljungbox
import scipy.stats

# ignore warning
import warnings
warnings.filterwarnings('ignore')
```

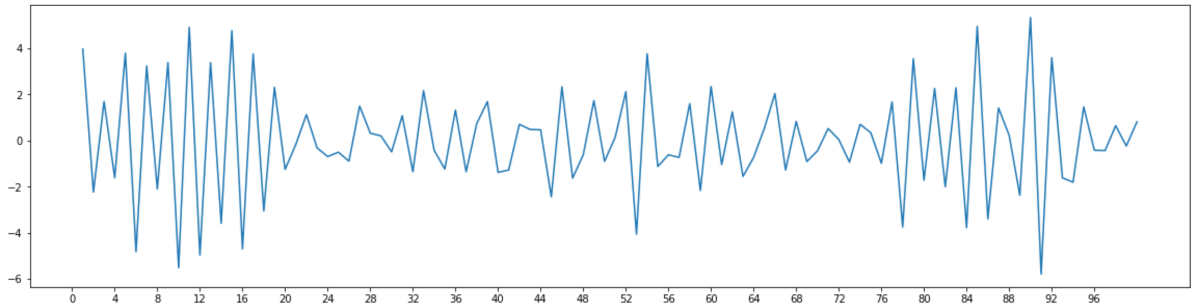
In [2]:
```
sim_2 = pd.read_excel('sim_2.xlsx')
NYSE = pd.read_excel('NYSE.xlsx')
```

## 1. The second column in the file SIM 2.XLSX contains the 100 values of the simulated ARMA(1,1) process. This series is entitled Y2. Use this series to perform the following tasks (Note: Due to differences in data handling and rounding, your answers need only approximate those presented here.):

### 1.a Plot the sequence against time. Does this series appear to be stationary?

In [3]:
```
plt.figure(figsize=(20,5))
plt.xticks(np.arange(0,99,4))
plt.plot(sim_2['OBS'],sim_2['Y2'])
```

Out[3]:
```
[<matplotlib.lines.Line2D at 0x7fa115380cd0>]
```




- Stationary process는 graph를 보고 시각적으로 확인이 어렵다.
- 따라서 time-series data가 stationary process인지 test하는 **ADF test**와 **KPSS test**를 진행한다.

## Check Stationary process

### ADF test

Null Hypotesis : Stationarity하지 않다.
Alternative Hypotesis : Stationarity하다.

In [4]:
```
# ADF test function
def adf_test(df):
    result  = adfuller(df.values)
    print('ADF Statistics: %f' % result[0])
    print('p-value: %f' % result[1])
    print('Critical value:')
    for key, value in result[4].items():
        print('\t%s: %.3f' % (key,value))

adf_test(sim_2['Y2'])
```

```
ADF Statistics: -8.057094
p-value: 0.000000
Critical value:
	1%: -3.501
	5%: -2.892
	10%: -2.583
```

- significance level 1%, 5%, 10%의 Critical value보다 ADF Statistics 값이 작기 때문에 Null Hypotesis를 reject할 수 있다.
- p-value가 매우 작아 0에 가깝다.

따라서 해당 데이터는 Null Hypotesis를 기각하기 때문에 stationary process를 만족한다.

### KPSS_test

Null Hypotesis : Stationarity하다.
Alternative Hypotesis : Stationarity하지 않다.

In [5]:
```
# KPSS test function
def kpss_test(df):
    statistic, p_value, n_lags, critical_values = kpss(df.values)

    print(f'KPSS Statistic: {statistic}')
    print(f'p-value: {p_value}')
```

```
    print(f'num lags: {n_lags}')
    print('Critical value:')

    for key, value in critical_values.items():
        print(f'{key} : {value}')

kpss_test(sim_2['Y2'])
```

```
KPSS Statistic: 0.04390055049378627
p-value: 0.1
num lags: 6
Critical value:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739
```

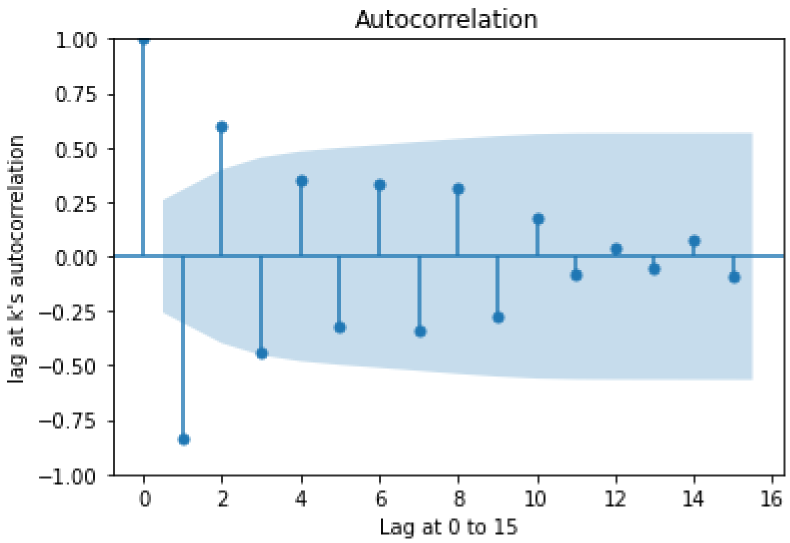
- significance level 1%, 2.5%, 5%, 10%의 Critical values에서 ADF Statistics 값이 기각역에 속하지 않기 때문에 Null Hypotesis를 reject하지 못한다.
- p-value가 0.1로 significance level에서 유의하지 않다.

따라서 해당 데이터는 Null Hypotesis를 기각하지 못하기 때문에 stationary process를 만족한다.

### 1.b Plot the ACF.

In [6]:
```
# plot acf chart function
def acf_plot(data, N_LAGS, alpha):
    fig = plot_acf(data, lags=N_LAGS, alpha=alpha)
    plt.xlabel(f'Lag at 0 to {N_LAGS}')
    plt.ylabel("lag at k's autocorrelation")
    plt.show()

# set Lags 15 and set significance level 0.01
acf_plot(sim_2['Y2'], 15, 0.01)
```




- ACF가 significance level 0.01하에서 lags 3에서 절단값을 가지므로 MA(2) model 생성

### 1.c Estimate the process using a pure MA(2) model. You should obtain

Observations: 100

$$y_t = -1.15(-13.22)\varepsilon_{t-1} + 0.522(5.98)\varepsilon_{t-2} + e_t$$

Where numbers in parentheses are t-statistics. Verify that the Ljung-Box Q-Statistics are Q(8) = 28.48, Q(16) = 37.47, and Q(24) = 38.84 with significance levels of 0.000, 0.000, and 0.015, respectively. Is this MA(2) model is a good model for explaining this sequence Y2? Explain.

In [7]:

```
# Create MA(2) model
model = sm.tsa.arima.ARIMA(sim_2['Y2'], order = (0,0,2), trend = 'n') # cons
MA2_result = model.fit()
MA2_result.summary()
```

Out[7]:

SARIMAX Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Y2 | **No. Observations:** | 100 |
| **Model:** | ARIMA(0, 0, 2) | **Log Likelihood** | -162.643 |
| **Date:** | Sun, 16 Apr 2023 | **AIC** | 331.286 |
| **Time:** | 16:03:28 | **BIC** | 339.102 |
| **Sample:** | 0 | **HQIC** | 334.449 |
| | - 100 | | |
| **Covariance Type:** | opg | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **ma.L1** | -1.2601 | 0.091 | -13.862 | 0.000 | -1.438 | -1.082 |
| **ma.L2** | 0.5517 | 0.095 | 5.809 | 0.000 | 0.366 | 0.738 |
| **sigma2** | 1.4873 | 0.238 | 6.248 | 0.000 | 1.021 | 1.954 |

| | | | |
|---|---|---|---|
| **Ljung-Box (L1) (Q):** | 2.04 | **Jarque-Bera (JB):** | 0.40 |
| **Prob(Q):** | 0.15 | **Prob(JB):** | 0.82 |
| **Heteroskedasticity (H):** | 0.74 | **Skew:** | -0.02 |
| **Prob(H) (two-sided):** | 0.40 | **Kurtosis:** | 2.69 |

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

- 생성한 MA(2) model에서 intercept의 p-value가 매우 크므로 insignificant하기 때문에 제거를 하고 MA(2) model를 재생성했다.
- AIC 값이 331.286이며, BIC 값은 339.102이다.
- 생성 결과 ma.L1, ma.L2 coef의 p-value가 매우 작으므로 significant하다.

하지만 MA(2) model에 대한 Ljung-Box test의 Q값을 Lags별로 확인하기 위해 MA(2) model에 대해 따로 Ljung-Box test를 실시한다.

## LjungBox test(MA(2))

Null Hypotesis : 자기상관계수가 0이다 (자기상관이 없다) Alternative Hypotesis : 자기상관계수가 0이 아니다 (자기상관이 있다)

In [8]:
```
# ljungbox test function
def ljungbox(data, N_LAGS):
    print(sm.stats.acorr_ljungbox(data.resid, lags=[N_LAGS]))

ljungbox(MA2_result, 1)
ljungbox(MA2_result, 2)
ljungbox(MA2_result, 8)
ljungbox(MA2_result, 16)
ljungbox(MA2_result, 24)
```

```
   lb_stat  lb_pvalue
1  1.59758   0.206247
     lb_stat  lb_pvalue
2  11.822506   0.002709
     lb_stat  lb_pvalue
8  30.094468   0.000203
      lb_stat  lb_pvalue
16  41.188217   0.000521
      lb_stat  lb_pvalue
24  43.074868   0.009753
```

- LjungBox test 결과 Lags가 1일 때 p-value가 0.206로 Null Hypotesis를 reject하지 못한다.
- 하지만 Lags 2부터 Q 값은 급격하게 커지며 p-value 또한 매우 작아져 Null Hypotesis를 reject 한다.

따라서 MA(2) model은 Y2를 explain하는데 good model이 아니다.

### 1.d Compare the MA(2) to the ARMA(1, 1).

In [9]:
```
# Create ARMA(1,1) model
model = sm.tsa.arima.ARIMA(sim_2['Y2'], order = (1,0,1), trend = 'n') # cons
ARMA11_result = model.fit()
ARMA11_result.summary()
```

Out[9]:

SARIMAX Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Y2 | **No. Observations:** | 100 |
| **Model:** | ARIMA(1, 0, 1) | **Log Likelihood** | -153.152 |
| **Date:** | Sun, 16 Apr 2023 | **AIC** | 312.304 |
| **Time:** | 16:03:28 | **BIC** | 320.119 |
| **Sample:** | 0 | **HQIC** | 315.467 |
| | - 100 | | |
| **Covariance Type:** | opg | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **ar.L1** | -0.7086 | 0.085 | -8.354 | 0.000 | -0.875 | -0.542 |
| **ma.L1** | -0.6649 | 0.093 | -7.155 | 0.000 | -0.847 | -0.483 |
| **sigma2** | 1.2270 | 0.199 | 6.181 | 0.000 | 0.838 | 1.616 |

| | | | |
|---|---|---|---|
| **Ljung-Box (L1) (Q):** | 0.26 | **Jarque-Bera (JB):** | 0.60 |
| **Prob(Q):** | 0.61 | **Prob(JB):** | 0.74 |
| **Heteroskedasticity (H):** | 0.93 | **Skew:** | 0.05 |
| **Prob(H) (two-sided):** | 0.84 | **Kurtosis:** | 2.63 |

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

- 생성한 ARMA(1,1) model에서 intercept의 p-value가 매우 크므로 insignificant하기 때문에 제거를 하고 ARMA(1,1) model를 재생성했다.
- AIC 값이 312.304이며, BIC 값은 320.119이다.
- 생성 결과 ar.L1, ma.L1 coef의 p-value가 매우 작으므로 significant하다.

또한, ARMA(1,1) model에 대한 Ljung-Box test의 Q값을 Lags별로 확인하기 위해 ARMA(1,1) model에 대해 따로 Ljung-Box test를 실시한다.

## LjungBox test(ARMA(1,1))

Null Hypotesis : 자기상관계수가 0이다 (자기상관이 없다)
Alternative Hypotesis : 자기상관계수가 0이 아니다 (자기상관이 있다)

In [10]:

```
ljungbox(ARMA11_result, 1)
ljungbox(ARMA11_result, 2)
ljungbox(ARMA11_result, 8)
ljungbox(ARMA11_result, 16)
ljungbox(ARMA11_result, 24)
ljungbox(ARMA11_result, 32)
ljungbox(ARMA11_result, 92)
ljungbox(ARMA11_result, 93)
```

```
    lb_stat  lb_pvalue
1  0.458335   0.498403
    lb_stat  lb_pvalue
2  0.478544   0.787201
    lb_stat  lb_pvalue
8  2.577774   0.958005
      lb_stat  lb_pvalue
16  12.897498   0.680237
      lb_stat  lb_pvalue
24  16.360301   0.874634
      lb_stat  lb_pvalue
32  24.979013   0.806878
       lb_stat  lb_pvalue
92  102.693509   0.209423
       lb_stat  lb_pvalue
93  117.417465   0.044383
```

### 비교 결과

- MA(2) model과 ARMA(1,1) model에서 intercept를 제외한 coef는 significant하다.
- ARMA(1,1) model이 MA(2) model보다 AIC와 BIC가 작고, Log Likelihood 값이 크므로 ARMA(1,1) model이 good model이다.
- LjungBox test 결과 ARMA(1,1) model은 Lags 92번째까지 데이터가 독립적이지만, MA(2) model은 1번째까지 데이터가 독립적이다.

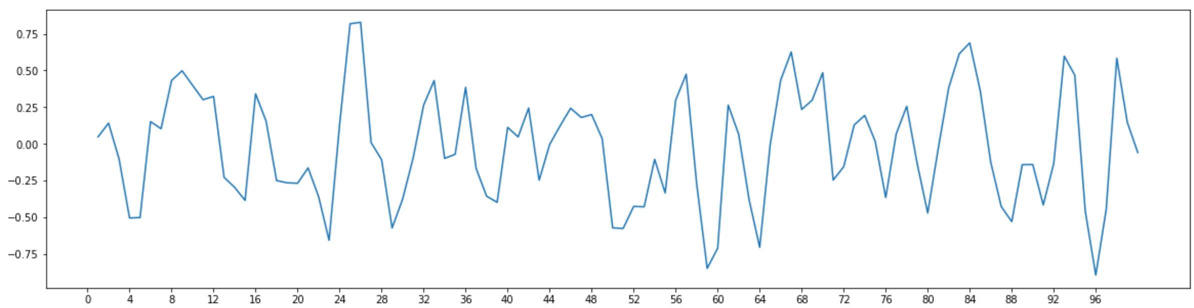
이러한 이유로 ARMA(1,1) model이 Y2를 explain하는 데 적합한 model이라고 할 수 있다.

## 2. The third column in file SIM 2.XLSX contains the 100 values of the simulated AR(2) process. This series is entitled Y3. Use this series to perform the follwing tasks.

### 1.a Plot the sequence against time. Show the ACF and PACF coefficients. Compare the sample ACF and PACF to those of a theoretical AR(2) process.

In [11]:
```
plt.figure(figsize=(20,5))
plt.xticks(np.arange(0,99,4))
plt.plot(sim_2['OBS'],sim_2['Y3'])
```

Out[11]:
```
[<matplotlib.lines.Line2D at 0x7fa115553130>]
```




### Check Stationary process

### ADF test

Null Hypotesis : Stationarity하지 않다.
Alternative Hypotesis : Stationarity하다.

In [12]:
```
adf_test(sim_2['Y3'])
```

```
ADF Statistics: -4.148853
p-value: 0.000803
Critical value:
        1%: -3.503
        5%: -2.893
        10%: -2.584
```

ADF test 실시 결과 p-value는 매우 작으므로 Null Hypotesis를 reject할 수 있다. 따라서 해당 data는 stationary process를 따른다.

## KPSS_test

Null Hypotesis : Stationarity하다.
Alternative Hypotesis : Stationarity하지 않다.

In [13]:
```
kpss_test(sim_2['Y3'])
```

```
KPSS Statistic: 0.03775377915722137
p-value: 0.1
num lags: 1
Critical value:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739
```

KPSS test 실시 결과 p-value는 크므로 Null Hypotesis를 reject할 수 없다. 따라서 해당 data는 stationary process를 따른다.

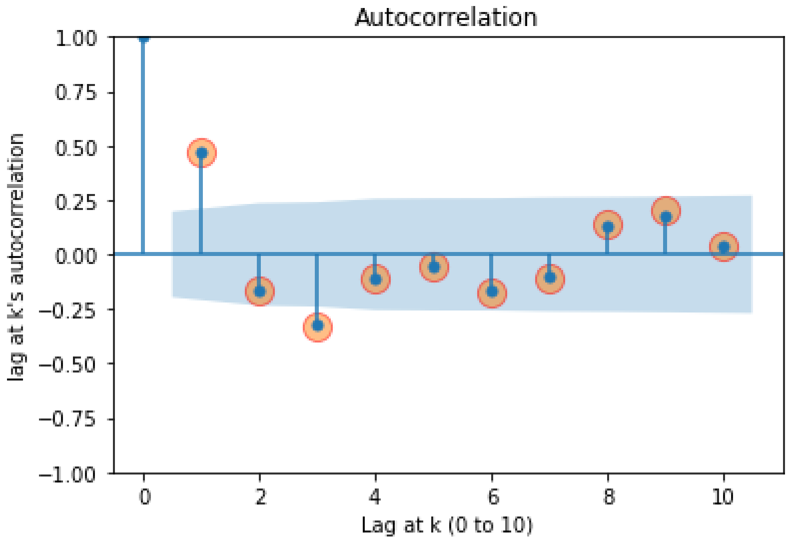
## ACF

In [14]:
```
def acf_plot_coef(data, N_LAGS, pval):
    auto = pd.Series(data.values)
    for i in range(0, N_LAGS+1):
        print(f"lag at {i}'s autocorrelation = ", round(auto.autocorr(lag=i)
        scatter = pd.DataFrame()
        scatter['lags'] = [i for i in range (1, N_LAGS +1)]
        scatter['autocorrelation'] = [ auto.autocorr(lag=i) for i in range(1

    fig = plot_acf(data, lags=N_LAGS, alpha=pval)
    plt.xlabel(f'Lag at k (0 to {N_LAGS})')
    plt.ylabel("lag at k's autocorrelation")
    plt.scatter(x=scatter['lags'], y=scatter['autocorrelation'], edgecolors=
    plt.show()

acf_plot_coef(sim_2['Y3'], 10, 0.05)
```

```
lag at 0's autocorrelation =  1.0
lag at 1's autocorrelation =  0.47
lag at 2's autocorrelation =  -0.16
lag at 3's autocorrelation =  -0.33
lag at 4's autocorrelation =  -0.11
lag at 5's autocorrelation =  -0.06
lag at 6's autocorrelation =  -0.18
lag at 7's autocorrelation =  -0.11
lag at 8's autocorrelation =  0.14
lag at 9's autocorrelation =  0.2
lag at 10's autocorrelation =  0.04
```




## PACF

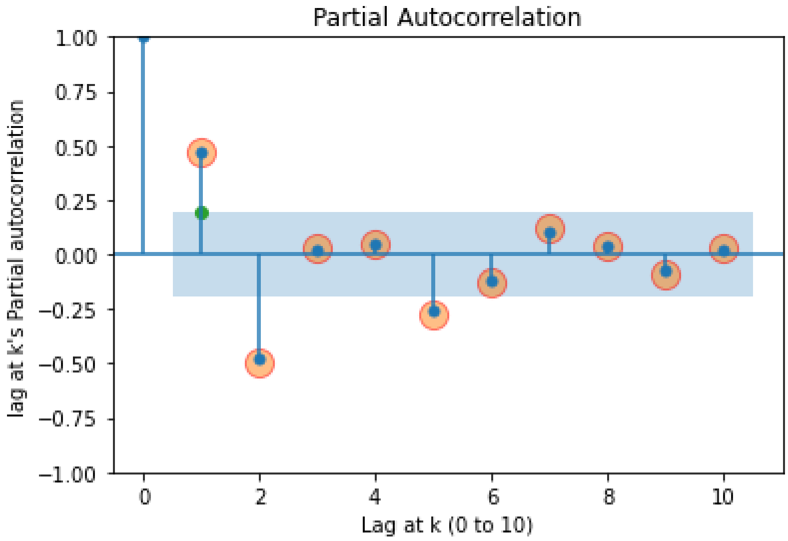
In [15]:
```
def pacf_plot_coef(data, N_LAGS, alpha):
    # 편자기상관계수를 구하는 부분
    auto = pd.Series(data.values)
    for i in range(0, N_LAGS+1):
        # lag 별 pacf 추정 계수를 출력하는 부분
        print(f"lag at {i}'s Partial autocorrelation = ", round(pacf(data, a
        scatter = pd.DataFrame()
        scatter['lags'] = [i for i in range (1, N_LAGS +1)]
        scatter['Partial autocorrelation'] = [pacf(data, alpha=0.05)[0][i] f

    print(f"1번째 lag에서 파란 음영의 값 범위는 -{scipy.stats.norm.ppf(1-(alpha)/2)

    # 표 그리는 부분
    plot_pacf(data, lags=N_LAGS, alpha=alpha, method='ywm')
    plt.xlabel(f'Lag at k (0 to {N_LAGS})')
    plt.ylabel("lag at k's Partial autocorrelation")
    # lag 별로 PACF 추정 계수를 점으로 찍는 부분
    plt.scatter(x=scatter['lags'], y=scatter['Partial autocorrelation'], edg
    # lag = 1 에서 신뢰구간의 upper 부분을 점으로 찍는 부분
    plt.scatter(x=1, y=[scipy.stats.norm.ppf(1-(alpha)/2) * (1/np.sqrt(data.
    plt.show()

pacf_plot_coef(sim_2['Y3'], 10, 0.05)
```

```
lag at 0's Partial autocorrelation =  1.0
lag at 1's Partial autocorrelation =  0.47
lag at 2's Partial autocorrelation =  -0.49
lag at 3's Partial autocorrelation =  0.03
lag at 4's Partial autocorrelation =  0.05
lag at 5's Partial autocorrelation =  -0.27
lag at 6's Partial autocorrelation =  -0.13
lag at 7's Partial autocorrelation =  0.12
lag at 8's Partial autocorrelation =  0.04
lag at 9's Partial autocorrelation =  -0.09
lag at 10's Partial autocorrelation =  0.03
1번째 lag에서 파란 음영의 값 범위는 -0.1969837921008876, +0.1969837921008876입니다.
```




- ACF가 2에서 절단점을 가지지만 3에서 다시 올라오고 4에서 절단점을 가진다.
- PACF가 3에서 절단점을 가진다.

따라서 ACF와 PACF 그래프를 통해 AR(2), MA(1), MA(3), ARMA(2,0,3) model을 추측할 수 있다.

## 2.b Estimate a series as an AR(1) process. You should find that the estimated AR(1) coefficient and the t-statistic in parentheses are

$$y_t = 0.467(5.24)y_{t-1} + e_t$$

## Show that the standard diagnostic checks indicate that this AR(1) model is inadequate.

In [16]:
```
# Create AR(1) model
model = sm.tsa.arima.ARIMA(sim_2['Y3'], order = (1,0,0), trend = 'n') # cons
AR1_result = model.fit()
AR1_result.summary()
```

Out[16]:

SARIMAX Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Y3 | **No. Observations:** | 100 |
| **Model:** | ARIMA(1, 0, 0) | **Log Likelihood** | -32.891 |
| **Date:** | Sun, 16 Apr 2023 | **AIC** | 69.782 |
| **Time:** | 16:03:29 | **BIC** | 74.993 |
| **Sample:** | 0 | **HQIC** | 71.891 |
| | - 100 | | |
| **Covariance Type:** | opg | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **ar.L1** | 0.4631 | 0.087 | 5.337 | 0.000 | 0.293 | 0.633 |
| **sigma2** | 0.1128 | 0.020 | 5.763 | 0.000 | 0.074 | 0.151 |

| | | | |
|---|---|---|---|
| **Ljung-Box (L1) (Q):** | 5.24 | **Jarque-Bera (JB):** | 1.85 |
| **Prob(Q):** | 0.02 | **Prob(JB):** | 0.40 |
| **Heteroskedasticity (H):** | 1.19 | **Skew:** | 0.10 |
| **Prob(H) (two-sided):** | 0.63 | **Kurtosis:** | 2.36 |

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

## LjungBox test(AR(1))

Null Hypotesis : 자기상관계수가 0이다 (자기상관이 없다)
Alternative Hypotesis : 자기상관계수가 0이 아니다 (자기상관이 있다)

In [17]:

```
ljungbox(AR1_result, 1)
ljungbox(AR1_result, 2)
ljungbox(AR1_result, 8)
ljungbox(AR1_result, 16)
```

```
    lb_stat  lb_pvalue
1  5.245152   0.022008
     lb_stat  lb_pvalue
2  16.855564   0.000219
     lb_stat  lb_pvalue
8  37.516164   0.000009
      lb_stat  lb_pvalue
16  56.642911   0.000002
```

- AR(1) model 결과 AIC 값은 69.782, BIC 값은 74.993이다.
- intercept를 제외한 coef는 p-value가 매우 작아 significant하다.
- Ljungbox test 결과 귀무가설을 기각하여 자기상관이 있는 것으로 확인되었다.

따라서 AR(1) model은 inadequate하다.

## auto_arima 함수를 사용해 최적의 ARIMA model 찾기

In [18]:

```
model = pm.auto_arima(y = sim_2['Y3']
                      , start_p = 0
                      , max_p = 5
                      , start_q = 0
                      , max_q = 5
                      , m = 1
                      , seasonal = False
                      , stepwise = True
                      , trace=True
                      )
```

```
Performing stepwise search to minimize aic
 ARIMA(0,0,0)(0,0,0)[0]             : AIC=92.210, Time=0.02 sec
 ARIMA(1,0,0)(0,0,0)[0]             : AIC=69.782, Time=0.02 sec
 ARIMA(0,0,1)(0,0,0)[0]             : AIC=56.159, Time=0.02 sec
 ARIMA(1,0,1)(0,0,0)[0]             : AIC=56.436, Time=0.03 sec
 ARIMA(0,0,2)(0,0,0)[0]             : AIC=52.437, Time=0.03 sec
 ARIMA(1,0,2)(0,0,0)[0]             : AIC=52.898, Time=0.05 sec
 ARIMA(0,0,3)(0,0,0)[0]             : AIC=49.403, Time=0.06 sec
 ARIMA(1,0,3)(0,0,0)[0]             : AIC=46.792, Time=0.09 sec
 ARIMA(2,0,3)(0,0,0)[0]             : AIC=41.496, Time=0.25 sec
 ARIMA(2,0,2)(0,0,0)[0]             : AIC=43.529, Time=0.07 sec
 ARIMA(3,0,3)(0,0,0)[0]             : AIC=45.702, Time=0.21 sec
 ARIMA(2,0,4)(0,0,0)[0]             : AIC=43.148, Time=0.30 sec
 ARIMA(1,0,4)(0,0,0)[0]             : AIC=46.072, Time=0.16 sec
 ARIMA(3,0,2)(0,0,0)[0]             : AIC=43.088, Time=0.19 sec
 ARIMA(3,0,4)(0,0,0)[0]             : AIC=42.636, Time=0.38 sec
 ARIMA(2,0,3)(0,0,0)[0] intercept   : AIC=43.244, Time=0.22 sec

Best model:  ARIMA(2,0,3)(0,0,0)[0]
Total fit time: 2.118 seconds
```

- auto_arima 결과 ARIMA(2,0,3)이 best model이라는 결과를 얻었다.

In [19]:

```
# Create ARNA(2,3) model
model = sm.tsa.arima.ARIMA(sim_2['Y3'], order = (2,0,3), trend = 'n') # cons
ARMA23_result = model.fit()
ARMA23_result.summary()
```

Out[19]:

SARIMAX Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Y3 | **No. Observations:** | 100 |
| **Model:** | ARIMA(2, 0, 3) | **Log Likelihood** | -14.748 |
| **Date:** | Sun, 16 Apr 2023 | **AIC** | 41.496 |
| **Time:** | 16:03:31 | **BIC** | 57.127 |
| **Sample:** | 0 | **HQIC** | 47.822 |
| | - 100 | | |
| **Covariance Type:** | opg | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **ar.L1** | 0.0135 | 0.088 | 0.153 | 0.879 | -0.160 | 0.187 |
| **ar.L2** | -0.8331 | 0.072 | -11.594 | 0.000 | -0.974 | -0.692 |
| **ma.L1** | 0.7143 | 0.136 | 5.267 | 0.000 | 0.449 | 0.980 |
| **ma.L2** | 1.0269 | 0.091 | 11.315 | 0.000 | 0.849 | 1.205 |
| **ma.L3** | 0.3627 | 0.123 | 2.960 | 0.003 | 0.123 | 0.603 |
| **sigma2** | 0.0768 | 0.014 | 5.440 | 0.000 | 0.049 | 0.104 |

| | | | |
|---|---|---|---|
| **Ljung-Box (L1) (Q):** | 0.03 | **Jarque-Bera (JB):** | 1.95 |
| **Prob(Q):** | 0.86 | **Prob(JB):** | 0.38 |
| **Heteroskedasticity (H):** | 0.78 | **Skew:** | 0.10 |
| **Prob(H) (two-sided):** | 0.49 | **Kurtosis:** | 2.34 |

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

## LjungBox test(ARMA(2,3))

Null Hypotesis : 자기상관계수가 0이다 (자기상관이 없다)
Alternative Hypotesis : 자기상관계수가 0이 아니다 (자기상관이 있다)

In [20]:

```
ljungbox(ARMA23_result, 1)
ljungbox(ARMA23_result, 2)
ljungbox(ARMA23_result, 8)
ljungbox(ARMA23_result, 16)
```

```
    lb_stat  lb_pvalue
1  0.036226   0.849049
    lb_stat  lb_pvalue
2  0.057367   0.971724
    lb_stat  lb_pvalue
8  6.185298   0.626484
     lb_stat  lb_pvalue
16  20.011246   0.219715
```

- AR(1) model과 비교하여 AIC, BIC 값이 작고, log-likelihood 값이 크다.
- ar.L1을 제외하고 모두 p-value가 매우 작아 significant하다.
- Ljungbox test 결과 귀무가설을 reject하지 못해 자기상관이 없다는 것을 확인했다.

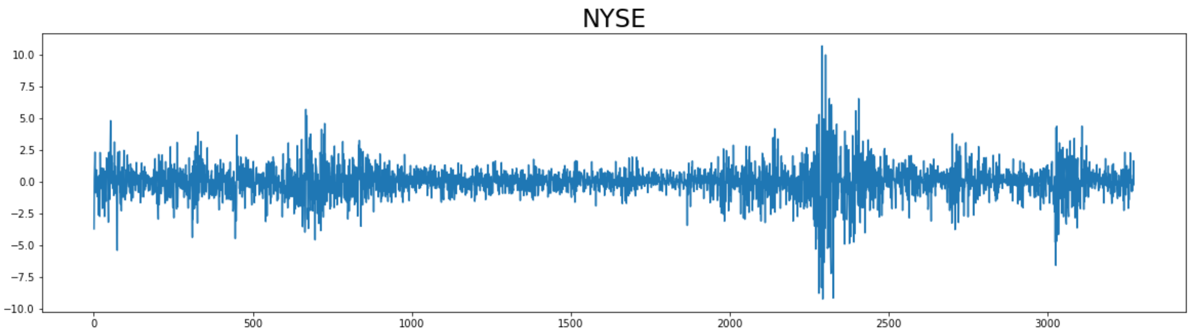
## 3. [bonus points] The file labeled NYSE.XLSX contains the daily values of the New York Stock Exchange Index.

In [21]:
```
df = NYSE[1:]
df.head()
```

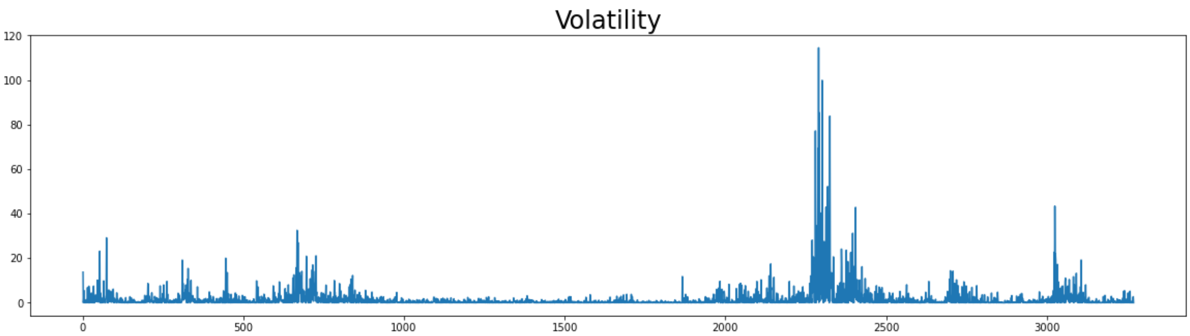
Out[21]:

| | Date | RETURN | RATE |
|---|---|---|---|
| 1 | 1900-03-23 | 7370.64 | -3.708476 |
| 2 | 1900-03-23 | 7387.27 | 0.225371 |
| 3 | 1900-03-23 | 7476.86 | 1.205467 |
| 4 | 1900-03-23 | 7653.59 | 2.336190 |
| 5 | 1900-03-23 | 7704.07 | 0.657394 |

In [22]:
```
df['RATE'].plot(figsize=(20,5))
plt.title("NYSE", size = 24)
plt.show()
```




In [23]:
```
df['sq_rates'] = df['RATE'].mul(df['RATE'])
df['sq_rates'].plot(figsize=(20,5))
plt.title("Volatility", size = 24)
plt.show()
```




## staionarity process 검증

In [24]:
```
adf_test(df['RATE'])
```

```
ADF Statistics: -13.606514
p-value: 0.000000
Critical value:
        1%: -3.432
        5%: -2.862
        10%: -2.567
```

In [25]:
```
kpss_test(df['RATE'])
```

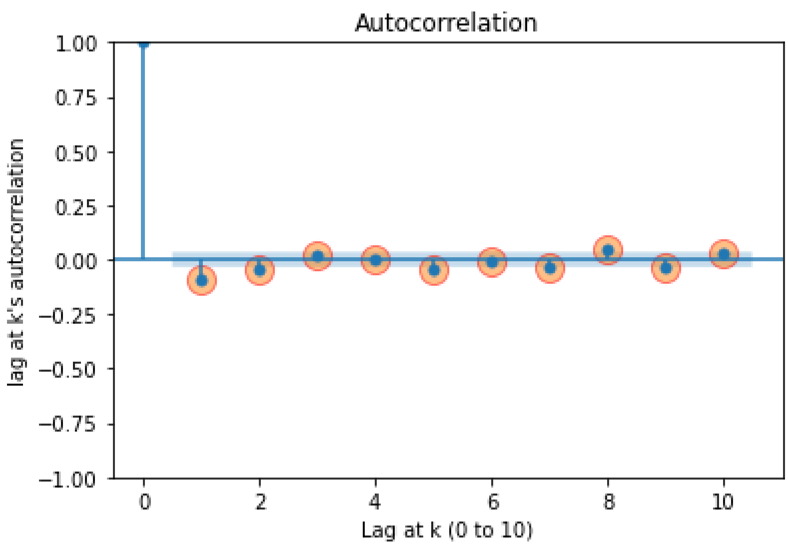
```
KPSS Statistic: 0.1049260326364128
p-value: 0.1
num lags: 20
Critical value:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739
```

- ADF test와 KPSS test 결과 해당 데이터는 stationarity process를 만족한다.

### 3.a Obtain the autocorrelations of the {rt} = 100 ∗ LN(Returnt/Returnt−1) (which is already computed on the column “RATE”). You may test the significance of ρ1 and ρ2.

In [26]:
```
acf_plot_coef(df['RATE'], 10, 0.05)
```

```
lag at 0's autocorrelation =  1.0
lag at 1's autocorrelation =  -0.09
lag at 2's autocorrelation =  -0.05
lag at 3's autocorrelation =  0.02
lag at 4's autocorrelation =  -0.0
lag at 5's autocorrelation =  -0.04
lag at 6's autocorrelation =  -0.01
lag at 7's autocorrelation =  -0.04
lag at 8's autocorrelation =  0.05
lag at 9's autocorrelation =  -0.03
lag at 10's autocorrelation =  0.02
```




## LjungBox test

Null Hypotesis : 자기상관계수가 0이다 (자기상관이 없다)

Alternative Hypotesis : 자기상관계수가 0이 아니다 (자기상관이 있다)

In [27]:
```
acorr_ljungbox(df['RATE'], lags=[1], return_df=True)
```

Out[27]:

| | lb_stat | lb_pvalue |
|---|---|---|
| 1 | 26.156327 | 3.148638e-07 |

In [28]:
```
acorr_ljungbox(df['RATE'], lags=[2], return_df=True)
```

Out[28]:

| | lb_stat | lb_pvalue |
|---|---|---|
| 2 | 34.012786 | 4.113555e-08 |

- LjungBox test 결과 자기상관이 있다고 확인되었다.

## 3.b Consider the AR(2) model estimated over the entire sample period

$$r_t = 0.0040(0.209) + \varepsilon_t - 0.0946(-5.42)r_{t-1} - 0.0575(-3.29)r_{t-2}$$

## Where numbers in parentheses are t-statistics. Is it possible to eliminate the intercept term from the regression? Explain your answer.

In [29]:
```
model = sm.tsa.arima.ARIMA(df['RATE'], order = (2,0,0))
result = model.fit()
result.summary()
```

Out[29]:

SARIMAX Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | RATE | **No. Observations:** | 3270 |
| **Model:** | ARIMA(2, 0, 0) | **Log Likelihood** | -5427.186 |
| **Date:** | Sun, 16 Apr 2023 | **AIC** | 10862.373 |
| **Time:** | 16:03:32 | **BIC** | 10886.743 |
| **Sample:** | 0 | **HQIC** | 10871.100 |
| | - 3270 | | |
| **Covariance Type:** | opg | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | 0.0031 | 0.020 | 0.157 | 0.875 | -0.036 | 0.042 |
| **ar.L1** | -0.0947 | 0.010 | -9.051 | 0.000 | -0.115 | -0.074 |
| **ar.L2** | -0.0576 | 0.008 | -7.018 | 0.000 | -0.074 | -0.041 |
| **sigma2** | 1.6185 | 0.019 | 85.679 | 0.000 | 1.581 | 1.656 |

| | | | |
|---|---|---|---|
| **Ljung-Box (L1) (Q):** | 0.00 | **Jarque-Bera (JB):** | 8572.58 |
| **Prob(Q):** | 0.97 | **Prob(JB):** | 0.00 |
| **Heteroskedasticity (H):** | 1.72 | **Skew:** | -0.36 |
| **Prob(H) (two-sided):** | 0.00 | **Kurtosis:** | 10.90 |

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

- intercept term의 p-value가 매우 크므로 해당 intercept는 insignificant하다. 따라서 intercept term 제거가 가능하다.

In [30]:
```
# intercept 제거한 model
model = sm.tsa.arima.ARIMA(df['RATE'], order = (2,0,0), trend = 'n')
AR2_result = model.fit()
AR2_result.summary()
```

Out[30]:

SARIMAX Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | RATE | **No. Observations:** | 3270 |
| **Model:** | ARIMA(2, 0, 0) | **Log Likelihood** | -5427.200 |
| **Date:** | Sun, 16 Apr 2023 | **AIC** | 10860.400 |
| **Time:** | 16:03:32 | **BIC** | 10878.677 |
| **Sample:** | 0 | **HQIC** | 10866.945 |
| | - 3270 | | |
| **Covariance Type:** | opg | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **ar.L1** | -0.0947 | 0.010 | -9.228 | 0.000 | -0.115 | -0.075 |
| **ar.L2** | -0.0576 | 0.008 | -7.112 | 0.000 | -0.073 | -0.042 |
| **sigma2** | 1.6185 | 0.019 | 86.815 | 0.000 | 1.582 | 1.655 |

| | | | |
|---|---|---|---|
| **Ljung-Box (L1) (Q):** | 0.00 | **Jarque-Bera (JB):** | 8572.78 |
| **Prob(Q):** | 0.97 | **Prob(JB):** | 0.00 |
| **Heteroskedasticity (H):** | 1.72 | **Skew:** | -0.36 |
| **Prob(H) (two-sided):** | 0.00 | **Kurtosis:** | 10.90 |

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

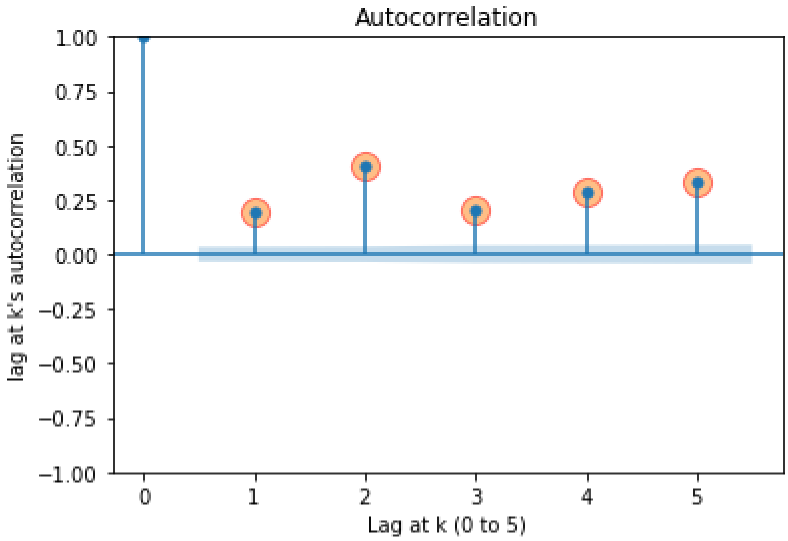
- intercept를 제거한 model이 AIC, BIC 값이 더 작다는 것을 확인할 수 있다.

## 3.c Obtain the ACF of the squared residuals for ρ1 to ρ5. The Q-statistics formed using the correlations of the suqared residuals are significant? If it is, the results imply strong evidence of GARCH errors. Explain your answer

In [31]:
```
from arch.univariate import ConstantMean,GARCH,EGARCH
from arch.univariate import Normal,StudentsT,GeneralizedError
from arch import arch_model
```

In [32]:
```
acf_plot_coef(AR2_result.resid**2, 5, 0.05)
```

```
lag at 0's autocorrelation =  1.0
lag at 1's autocorrelation =  0.2
lag at 2's autocorrelation =  0.41
lag at 3's autocorrelation =  0.2
lag at 4's autocorrelation =  0.29
lag at 5's autocorrelation =  0.33
```




## LjungBox test

Null Hypotesis : 자기상관계수가 0이다 (자기상관이 없다)

Alternative Hypotesis : 자기상관계수가 0이 아니다 (자기상관이 있다)

In [33]:
```
def ljungbox_square(data, N_LAGS):
    print(sm.stats.acorr_ljungbox(data.resid**2, lags=[N_LAGS]))
```

In [34]:
```
ljungbox_square(AR2_result, 1)
ljungbox_square(AR2_result, 2)
ljungbox_square(AR2_result, 3)
ljungbox_square(AR2_result, 4)
ljungbox_square(AR2_result, 5)
```

```
      lb_stat      lb_pvalue
1  127.588043  1.381351e-29
      lb_stat       lb_pvalue
2  671.261022  1.727910e-146
      lb_stat       lb_pvalue
3  803.038414  9.490129e-174
       lb_stat       lb_pvalue
4  1071.207062  1.318243e-230
       lb_stat       lb_pvalue
5  1429.714161  5.013238e-307
```

- Squared residual에 대한 Ljungbox test 결과 p-value가 매우 작아 Null Hypotesis를 기각한다.
- 따라서 squared residual에 autocorrelation이 있음을 알 수 있고, 이는 GARCH error의 strong evidence이다.

## 3.d Estimate the model of $r_t$ in 3.b using a GARCH(1,1) process.

In [35]:
```
garch11 =  arch_model(AR2_result.resid, p=1, q=1)
res = garch11.fit(update_freq=10)
print(res.summary())
```

```
Iteration:     10,   Func. Count:     62,   Neg. LLF: 4650.102762699566
Optimization terminated successfully    (Exit mode 0)
            Current function value: 4650.102762699566
            Iterations: 11
            Function evaluations: 66
            Gradient evaluations: 11
                     Constant Mean - GARCH Model Results
============================================================================
==
Dep. Variable:                      None   R-squared:                    0.0
00
Mean Model:                Constant Mean   Adj. R-squared:               0.0
00
Vol Model:                         GARCH   Log-Likelihood:             -4650.
10
Distribution:                     Normal   AIC:                         9308.
21
Method:               Maximum Likelihood   BIC:                         9332.
58
                                           No. Observations:               32
70
Date:                   Sun, Apr 16 2023   Df Residuals:                   32
69
Time:                           16:03:33   Df Model:
1
                                 Mean Model
============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
----------------------------------------------------------------------------
mu             0.0457  1.475e-02      3.099  1.942e-03 [1.680e-02,7.460e-02]
                              Volatility Model
============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
----------------------------------------------------------------------------
omega          0.0138  4.662e-03      2.960  3.079e-03 [4.661e-03,2.293e-02]
alpha[1]       0.0838  1.030e-02      8.136  4.085e-16   [6.359e-02,  0.104]
beta[1]        0.9063  1.063e-02     85.258      0.000     [  0.886,  0.927]
============================================================================

Covariance estimator: robust
```

# 출처

https://skyeong.net/285
https://blog.naver.com/PostView.nhn?
blogId=pmw9440&logNo=221709536663&parentCategoryNo=&categoryNo=7&viewDate=&
https://signature95.tistory.com/24
https://signature95.tistory.com/25
https://assaeunji.github.io/data%20analysis/2021-09-25-arimastock/