In [136]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pmdarima
from matplotlib.dates import MONDAY
from matplotlib.dates import WeekdayLocator
import statsmodels.tsa.arima.model as stm
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import ssl
from nelson_siegel_svensson import NelsonSiegelSvenssonCurve
from nelson_siegel_svensson.calibrate import calibrate_ns_ols
from nelson_siegel_svensson.calibrate import calibrate_nss_ols
import arch
import warnings

warnings.filterwarnings('ignore')
```

In [137]:
```
url_trates = "https://www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=yieldYear&year=2019"
url_gold_etf = "https://finance.yahoo.com/quote/GLD/history?period1=1569888000&period2=1575072000&interval=1d&filter=history&frequency=1d&includeAdjustec
url_equity_etf = "https://finance.yahoo.com/quote/CSUK.L/history?period1=1569888000&period2=1575072000&interval=1d&filter=history&frequency=1d&includeAdj
```

In [138]:
```
#Creating the interest rate data frame
dfs = pd.read_html(url_trates, header = 0, index_col=0, parse_dates = True)
df_trate = dfs[1]
df_trate = df_trate.loc['2019-10':'2019-11'].iloc[:,5:]
```

In [139]:
```
#Creating the gold ETF data frame
df_goldetf = pd.read_html(url_gold_etf, header = 0, index_col=0, parse_dates = True)[0].iloc[:-1]
df_goldetf.index = pd.to_datetime(df_goldetf.index)
df_goldetf = df_goldetf.sort_index(ascending = True)
df_goldetf = df_goldetf.astype(float)
```

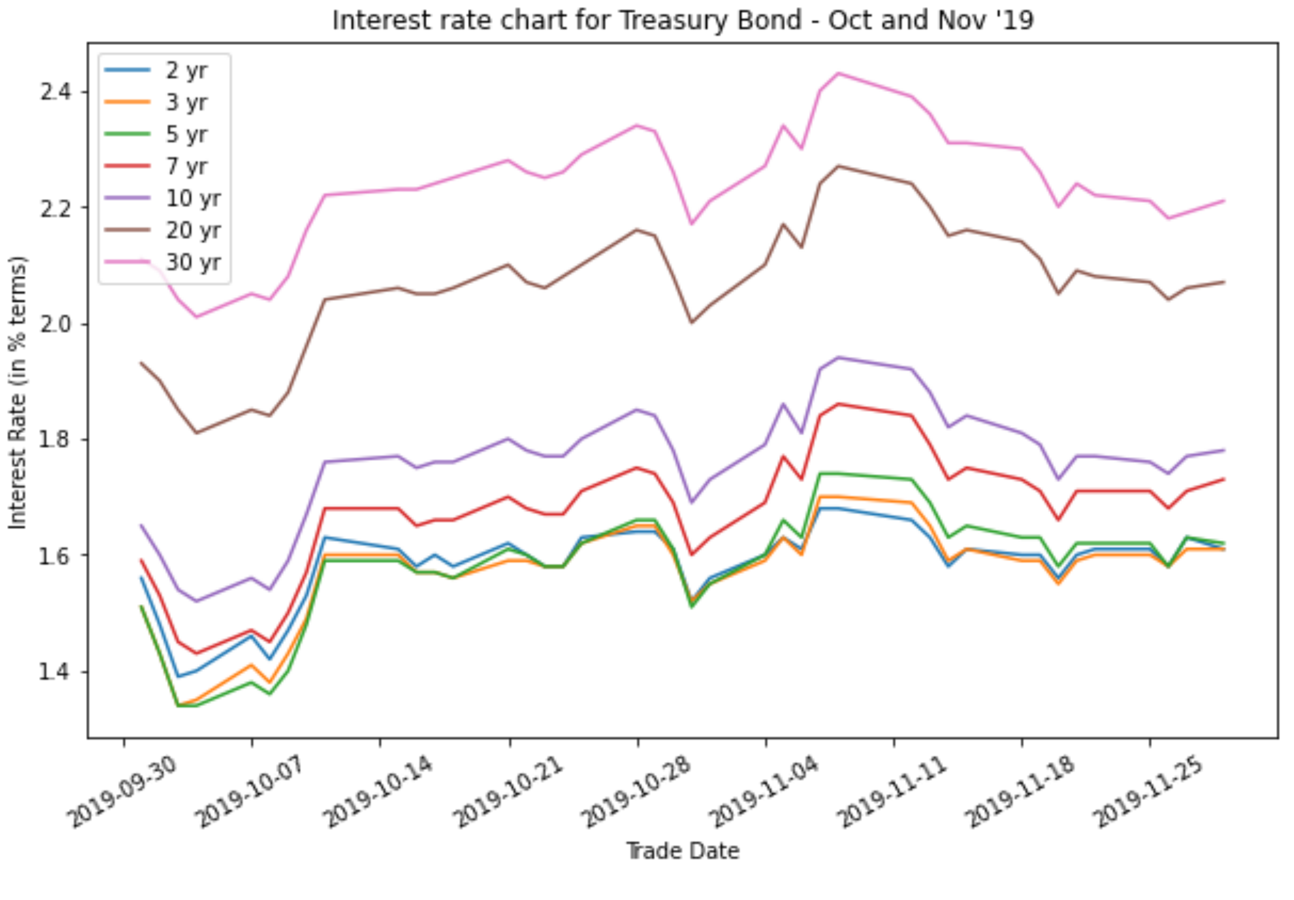
In [140]:
```
#Creating the equity ETF dataframe
df_equityetf = pd.read_html(url_equity_etf, header = 0, index_col=0)[0].iloc[:-1]
df_equityetf.index = pd.to_datetime(df_equityetf.index)
df_equityetf = df_equityetf.sort_index(ascending = True)
df_equityetf = df_equityetf.astype(float)
```

In [141]:
```
#Daily return on gold etf and equity etf
df_goldetf_daily_return = np.log(df_goldetf['Adj Close**']/df_goldetf['Adj Close**'].shift(1)).iloc[1:]
df_equityetf_daily_return = np.log(df_equityetf['Adj Close**']/df_equityetf['Adj Close**'].shift(1)).iloc[1:]
```

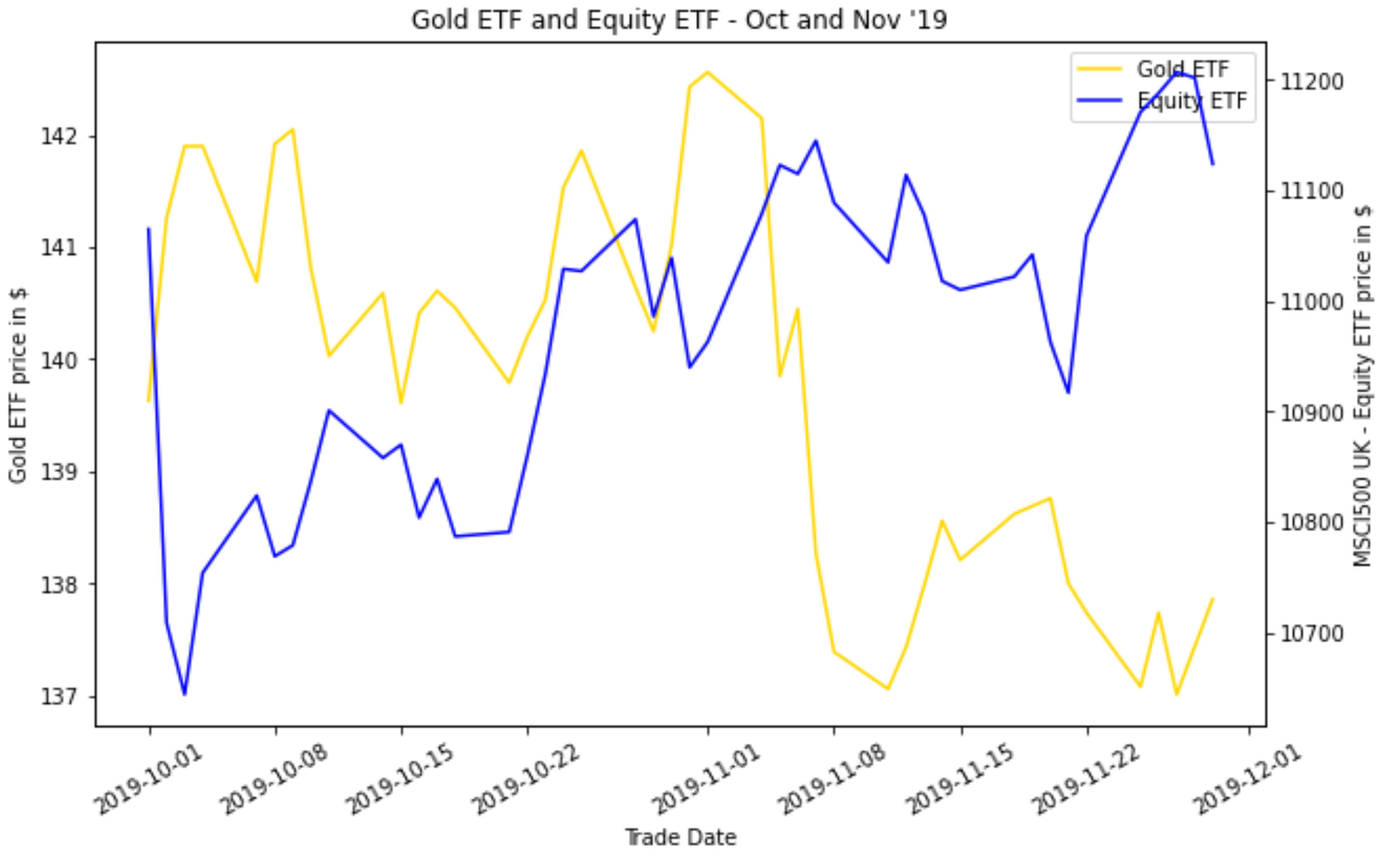
In [142]:
```
#Average return for yield for benchmark Security
df_avgmonthlyyield = df_trate.resample('M').mean()
#Average price of gold ETF on a monthly basis
df_avggoldETF = df_goldetf['Adj Close**'].resample('M').mean()
#Average price of equityETF on a monthly basis
df_avgequityETF = df_equityetf['Adj Close**'].resample('M').mean()
```

In [143]:
```
#Standard Deviation for yield of benchmark security
df_tratestd = df_trate.resample('M').std()
#Standard Deviation of gold ETF on a monthly basis
df_goldETFstd = df_goldetf['Adj Close**'].resample('M').std()
#Average price of equityETF on a monthly basis
df_equityETFstd = df_equityetf['Adj Close**'].resample('M').std()
```

In [144]:
```
#Plotting the yields
mondays = WeekdayLocator(MONDAY)
fig, ax = plt.subplots(figsize=(10,6))
ax.set_title("Interest rate chart for Treasury Bond - Oct and Nov '19")
ax.plot(df_trate)
ax.set_xlabel("Trade Date")
plt.xticks(rotation=30)
ax.xaxis.set_major_locator(mondays)
ax.set_ylabel("Interest Rate (in % terms)")
ax.legend(df_trate.columns.values)
plt.show()
```



In [145]:
```
#Plotting the gold and equity ETF
#mondays = WeekdayLocator(MONDAY)
fig, ax1 = plt.subplots(figsize=(10,6))
ax1.set_title("Gold ETF and Equity ETF - Oct and Nov '19")
lns1 = ax1.plot(df_goldetf['Adj Close**'], color = 'gold', label = "Gold ETF")
ax1.set_xlabel("Trade Date")
plt.xticks(rotation=30)
ax1.xaxis.set_major_locator(mondays)
ax1.set_ylabel("Gold ETF price in $")
ax2 = ax1.twinx()
ax2.set_ylabel("MSCI500 UK - Equity ETF price in $")
lns2 = ax2.plot(df_equityetf['Adj Close**'], color = 'blue', label = "Equity ETF")
lns = lns1+lns2
labs = [l.get_label() for l in lns]
ax1.legend(lns, labs)
plt.show()
```



In [146]:
```
ssl._create_default_https_context = ssl._create_unverified_context

#US Treasury Yield Data Frame
dfs = pd.read_html(url_trates, header = 0, index_col=0, parse_dates = True)
df_trate1 = dfs[1]
df_trate1 = df_trate1.loc['2019-10':'2019-11'].iloc[:,5:]
df_trate1 = df_trate1/100

#Nelson-Siegel model to fit the daily October and November 2019 yield curves
t = np.array([2, 3, 5, 7, 10, 20, 30])
yields = df_trate1.to_numpy()
df_nsyields = pd.DataFrame(columns = ['beta0', 'beta1', 'beta2', 'tau'])

for y in yields:
    curve1, status1 = calibrate_ns_ols(t, y)
    assert status1.success

    ns_elements = str(curve1).split(',')
    ns_beta0 = float(ns_elements[0].split('=')[1])
    ns_beta1 = float(ns_elements[1].split('=')[1])
    ns_beta2 = float(ns_elements[2].split('=')[1])
    ns_tau = float(ns_elements[3].split('=')[1][0:2])

    df_nsyields = df_nsyields.append({'beta0' : ns_beta0, 'beta1' : ns_beta1, 'beta2' : ns_beta2, 'tau' : ns_tau}, ignore_index = True)

print('Fit Based on Nelson-Siegel Model for Daily October and November 2019 Yield Curves')
print(df_nsyields)
print(" ")

#Nelson-Siegel-Svensson model to fit the daily October and November 2019 yield curves
df_nssyields = pd.DataFrame(columns = ['beta0', 'beta1', 'beta2', 'beta3', 'tau1', 'tau2'])

for y in yields:
    curve2, status2 = calibrate_nss_ols(t, y)
    assert status2.success

    nss_elements = str(curve2).split(',')
    nss_beta0 = float(nss_elements[0].split('=')[1])
    nss_beta1 = float(nss_elements[1].split('=')[1])
    nss_beta2 = float(nss_elements[2].split('=')[1])
    nss_beta3 = float(nss_elements[3].split('=')[1])
    nss_tau1 = float(nss_elements[4].split('=')[1])
    nss_tau2 = float(nss_elements[5].split('=')[1][0:2])

    df_nssyields = df_nssyields.append({'beta0' : nss_beta0, 'beta1' : nss_beta1, 'beta2' : nss_beta2, 'beta3' : nss_beta3, 'tau1' : nss_tau1, 'tau2' : r

print('Fit Based on Nelson-Siegel-Svensson Model for Daily October and November 2019 Yield Curves')
print(df_nssyields)
```

```
Fit Based on Nelson-Siegel Model for Daily October and November 2019 Yield Curves
       beta0     beta1     beta2  tau
0   0.022276  0.001521 -0.028021  2.0
1   0.022228  0.001185 -0.030047  2.0
2   0.021884  0.000904 -0.031485  2.0
3   0.021432  0.001603 -0.030968  2.0
4   0.021835  0.002271 -0.031698  2.0
5   0.021751  0.001701 -0.031478  2.0
6   0.022146  0.001780 -0.031295  2.0
7   0.022960  0.001259 -0.030932  2.0
8   0.023461  0.000960 -0.028354  2.0
9   0.023629  0.000334 -0.028094  2.0
10  0.023656  0.000239 -0.029052  2.0
11  0.023763  0.000597 -0.029757  2.0
12  0.023910  0.000153 -0.029940  2.0
13  0.024220 -0.000069 -0.029244  2.0
14  0.023896 -0.000097 -0.028469  2.0
15  0.023808 -0.000254 -0.028431  2.0
16  0.024003 -0.000205 -0.029235  2.0
17  0.024217  0.000137 -0.029053  2.0
18  0.024783 -0.000822 -0.028428  2.0
19  0.024654 -0.000615 -0.028389  2.0
20  0.023910 -0.000034 -0.028280  2.0
21  0.023082  0.000144 -0.028826  2.0
22  0.023456  0.000239 -0.029052  2.0
23  0.024143 -0.000300 -0.028944  2.0
24  0.024835 -0.001606 -0.027443  2.0
25  0.024414 -0.001007 -0.028026  2.0
26  0.025381 -0.002237 -0.025966  2.0
27  0.025775 -0.002604 -0.026631  2.0
28  0.025315 -0.002773 -0.025195  2.0
29  0.025027 -0.002297 -0.026373  2.0
30  0.024581 -0.001967 -0.027401  2.0
31  0.024587 -0.001775 -0.026888  2.0
32  0.024472 -0.001339 -0.027879  2.0
33  0.024001 -0.001016 -0.026846  2.0
34  0.023357 -0.000622 -0.026805  2.0
35  0.023740 -0.000769 -0.026438  2.0
36  0.023529 -0.000527 -0.025847  2.0
37  0.023384 -0.000466 -0.025441  2.0
38  0.023107 -0.000397 -0.025588  2.0
39  0.023170 -0.000175 -0.024709  2.0
40  0.023354 -0.000925 -0.024268  2.0

Fit Based on Nelson-Siegel-Svensson Model for Daily October and November 2019 Yield Curves
       beta0     beta1     beta2     beta3  tau1  tau2
0   0.025087 -0.006946 -0.008866 -0.018214   2.0   5.0
1   0.025014 -0.007208 -0.011060 -0.018055   2.0   5.0
2   0.024461 -0.006860 -0.013920 -0.016702   2.0   5.0
3   0.024255 -0.006901 -0.011730 -0.018293   2.0   5.0
4   0.024704 -0.006372 -0.012145 -0.018593   2.0   5.0
5   0.024744 -0.007317 -0.011078 -0.019398   2.0   5.0
6   0.024974 -0.006741 -0.012018 -0.018330   2.0   5.0
7   0.025772 -0.007214 -0.011764 -0.018226   2.0   5.0
8   0.026160 -0.007171 -0.009961 -0.017489   2.0   5.0
9   0.026324 -0.007785 -0.009727 -0.017465   2.0   5.0
10  0.026573 -0.008548 -0.009173 -0.018902   2.0   5.0
11  0.026486 -0.007604 -0.011204 -0.017641   2.0   5.0
12  0.026665 -0.008147 -0.011162 -0.017855   2.0   5.0
13  0.026829 -0.007927 -0.011466 -0.016905   2.0   5.0
14  0.026857 -0.009017 -0.008289 -0.019188   2.0   5.0
15  0.026752 -0.009121 -0.008370 -0.019074   2.0   5.0
16  0.026953 -0.009092 -0.009132 -0.019116   2.0   5.0
17  0.027244 -0.008983 -0.008421 -0.019618   2.0   5.0
18  0.027784 -0.009861 -0.007980 -0.019444   2.0   5.0
19  0.027769 -0.009999 -0.007162 -0.020185   2.0   5.0
20  0.026887 -0.009005 -0.007987 -0.019296   2.0   5.0
21  0.025998 -0.008641 -0.008952 -0.018897   2.0   5.0
22  0.026373 -0.008548 -0.009173 -0.018902   2.0   5.0
23  0.026855 -0.008470 -0.010460 -0.017575   2.0   5.0
24  0.027430 -0.009422 -0.009759 -0.016815   2.0   5.0
25  0.027207 -0.009421 -0.008991 -0.018100   2.0   5.0
26  0.028232 -0.010823 -0.006542 -0.018470   2.0   5.0
27  0.028470 -0.010724 -0.008263 -0.017465   2.0   5.0
28  0.027984 -0.010812 -0.007007 -0.017294   2.0   5.0
29  0.027793 -0.010630 -0.007521 -0.017926   2.0   5.0
30  0.027347 -0.010300 -0.008550 -0.017925   2.0   5.0
31  0.027065 -0.009239 -0.010001 -0.016057   2.0   5.0
32  0.027175 -0.009481 -0.009460 -0.017514   2.0   5.0
33  0.026718 -0.009201 -0.008331 -0.017606   2.0   5.0
34  0.026181 -0.009129 -0.007559 -0.018300   2.0   5.0
35  0.026534 -0.009185 -0.007398 -0.018104   2.0   5.0
36  0.026132 -0.008366 -0.008113 -0.016863   2.0   5.0
37  0.026070 -0.008559 -0.007134 -0.017408   2.0   5.0
38  0.025641 -0.008032 -0.008316 -0.016423   2.0   5.0
39  0.025525 -0.007268 -0.008661 -0.015259   2.0   5.0
40  0.025761 -0.008176 -0.007863 -0.015599   2.0   5.0
```

In [147]:

```
#Test for stationarity of the time series. We notice that the time series is non-stationary using ADF test
adfuller(df_goldetf_daily_return['Oct-2019'])
```

Out[147]:

```
(-2.6150837431841722,
 0.08991966550178265,
 9,
 12,
 {'1%': -4.137829282407408,
  '5%': -3.1549724074074077,
  '10%': -2.7144769444444443},
 -117.8033950462237)
```

In [148]:

```
#We differentiate the gold ETF daily returns and thus obtain a stationary series
d1_gold_etf_returns = df_goldetf_daily_return.diff().dropna()
#Proof to display stationarity of the Time Series - The first order differencing is causing the time series of gold ETF to be stationary
#Refer adf p-value for stationarity
adfuller(d1_gold_etf_returns['Oct-2019'])
```

Out[148]:

```
(-3.72735004523658,
 0.0037444905791674983,
 4,
 16,
 {'1%': -3.9240193847656246, '5%': -3.0684982031250003, '10%': -2.67389265625},
 -98.84113173699902)
```

In [149]:

```
plot_pacf(df_goldetf_daily_return['Oct-2019'],lags = 8);
plot_acf(df_goldetf_daily_return['Oct-2019'], lags = 8);
arimaoctgoldetfreturn = ARIMA(df_goldetf_daily_return['Oct-2019'], order = (1,0,0))
arimaoctgoldetfreturnfit = arimaoctgoldetfreturn.fit()
arimaoctgoldetfreturnfit.summary()
```

Out[149]:

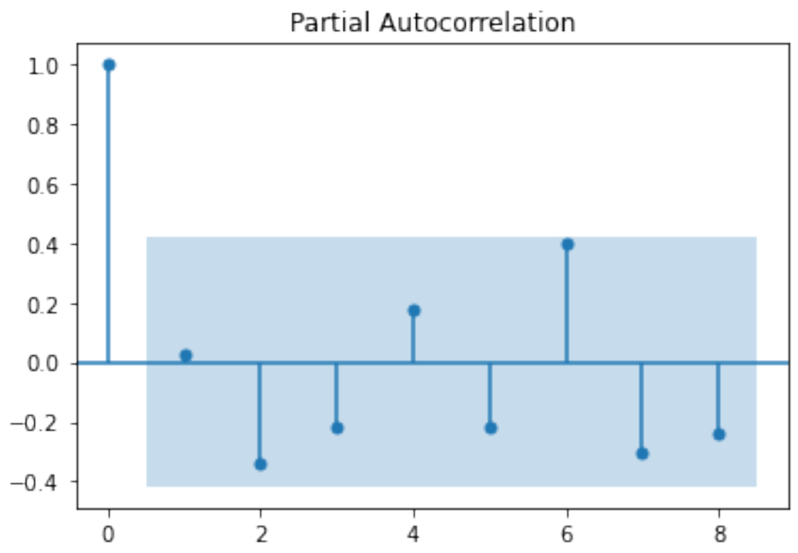
SARIMAX Results

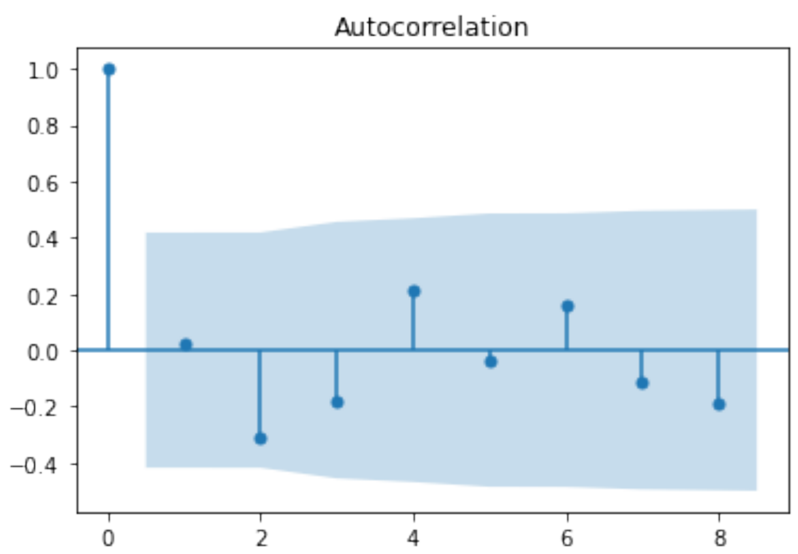
Dep. Variable:	Adj Close**	**No. Observations:**	22
Model:	ARIMA(1, 0, 0)	**Log Likelihood**	81.239
Date:	Tue, 12 Jan 2021	**AIC**	-156.477
Time:	14:11:59	**BIC**	-153.204
Sample:	10-02-2019	**HQIC**	-155.706
	- 10-31-2019		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0009	0.001	0.681	0.496	-0.002	0.004
ar.L1	0.0253	0.284	0.089	0.929	-0.532	0.582
sigma2	3.627e-05	1.56e-05	2.331	0.020	5.78e-06	6.68e-05

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	0.96
Prob(Q):	0.97	**Prob(JB):**	0.62
Heteroskedasticity (H):	0.66	**Skew:**	-0.13
Prob(H) (two-sided):	0.59	**Kurtosis:**	2.01

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

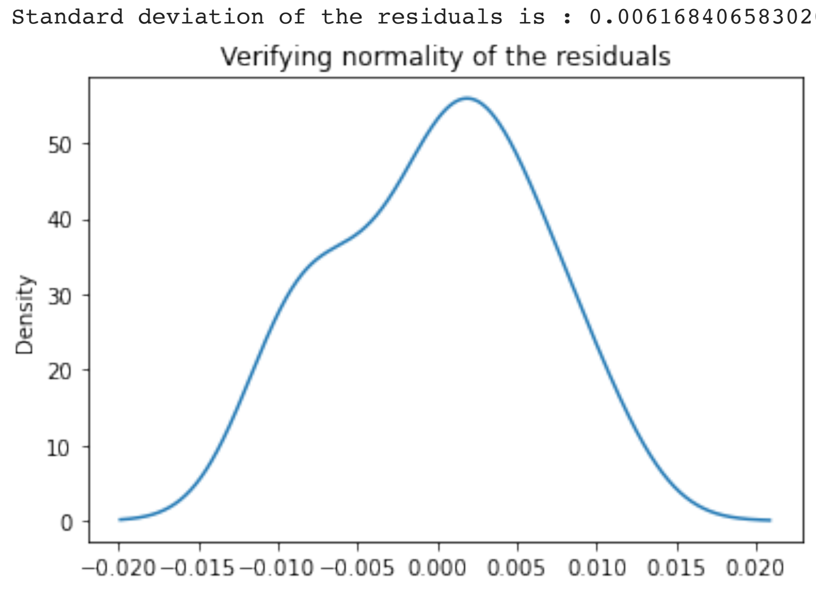




In [150]:

```
print('Standard deviation of the residuals is :',arimaoctgoldetfreturnfit.resid.std())
arimaoctgoldetfreturnfit.resid.plot(kind='kde', title = 'Verifying normality of the residuals');
```

```
Standard deviation of the residuals is : 0.006168406583020979
```



In [151]:

```
#We notice that the time series ARIMA(0,1,1)has the lowest AIC at -145 and the p-value of the MA co-efficient is less than the critical value.
#when we tried with a ARIMA(1,1,1) model, we had AIC of -143.15 and the p-value of the AR coefficient was 0.722 which is more than the critical value
```

In [152]:

```
#Proof to display stationarity of the Time Series - The 2nd order differencing is causing the time series of gold ETF to be stationary
#Refer adf p-value for stationarity
adfuller(df_goldetf_daily_return.diff().diff().dropna()['Nov-2019'])

plot_pacf(df_goldetf_daily_return.diff().diff().dropna()['Nov-2019'],lags = 8);
plot_acf(df_goldetf_daily_return.diff().diff().dropna()['Nov-2019'], lags = 8);

arimanovgoldetfreturn = ARIMA(df_goldetf_daily_return['Nov-2019'], order = (1,1,0))
arimanovgoldetfreturnfit = arimanovgoldetfreturn.fit()
arimanovgoldetfreturnfit.summary()
#We notice that the time series has the lowest AIC at -132.4 and the p-value of the AR co-efficient is less than the critical value.
#when we tried with a ARIMA(1,1,1) model, we had AIC of -129.9 and the p-value of the AR and MA were 0.394 and 0.539 which is more than the critical valu
```

Out[152]:

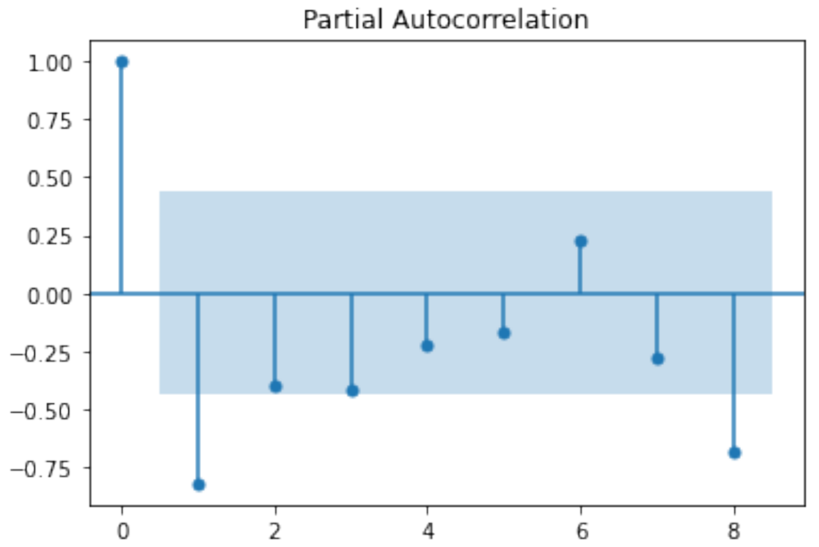
SARIMAX Results

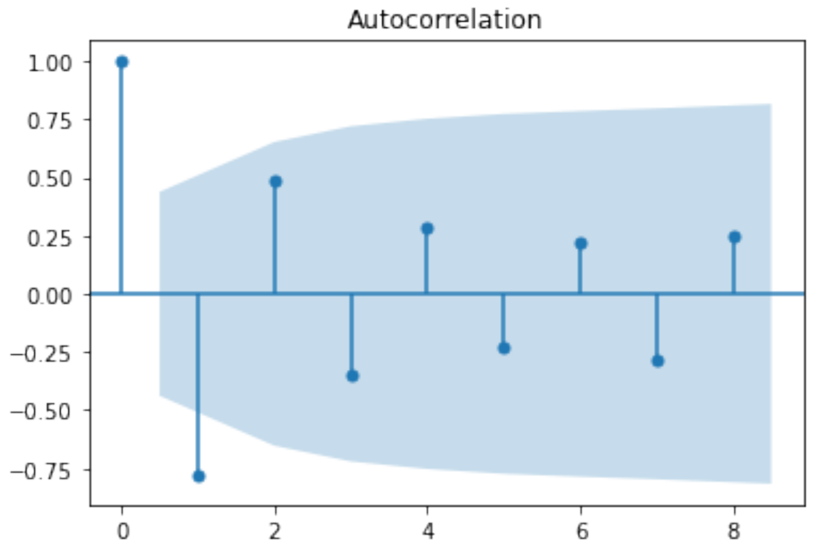
Dep. Variable:	Adj Close**	**No. Observations:**	20
Model:	ARIMA(1, 1, 0)	**Log Likelihood**	68.235
Date:	Tue, 12 Jan 2021	**AIC**	-132.470
Time:	14:12:01	**BIC**	-130.581
Sample:	0	**HQIC**	-132.151
	- 20		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.7316	0.171	-4.290	0.000	-1.066	-0.397
sigma2	4.267e-05	1.35e-05	3.153	0.002	1.61e-05	6.92e-05

Ljung-Box (L1) (Q):	0.48	**Jarque-Bera (JB):**	0.34
Prob(Q):	0.49	**Prob(JB):**	0.84
Heteroskedasticity (H):	0.21	**Skew:**	-0.32
Prob(H) (two-sided):	0.08	**Kurtosis:**	3.15

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





In [153]:

```
#Measuring the Equity ETF ARIMA
adfuller(df_equityetf_daily_return['Oct-2019'])
#We notice that the time series is stationary and the p-value is much less than the critical value which indicates that we reject the null hypothesis and
```

Out[153]:

```
(-3.9667390078920097,
 0.0015950474122178766,
 5,
 16,
 {'1%': -3.9240193847656246, '5%': -3.0684982031250003, '10%': -2.67389265625},
 -92.11804899346302)
```

In [154]:

```
from pandas.plotting import autocorrelation_plot
autocorrelation_plot(df_equityetf_daily_return['Oct-2019'])
plot_pacf(df_equityetf_daily_return['Oct-2019'],lags = 8);
plot_acf(df_equityetf_daily_return['Oct-2019'], lags = 10);
arimaoctequityreturn = ARIMA(df_equityetf_daily_return['Oct-2019'], order = (0,0,1))
arimaoctequityreturnfit = arimaoctequityreturn.fit()
arimaoctequityreturnfit.summary()
```

Out[154]:

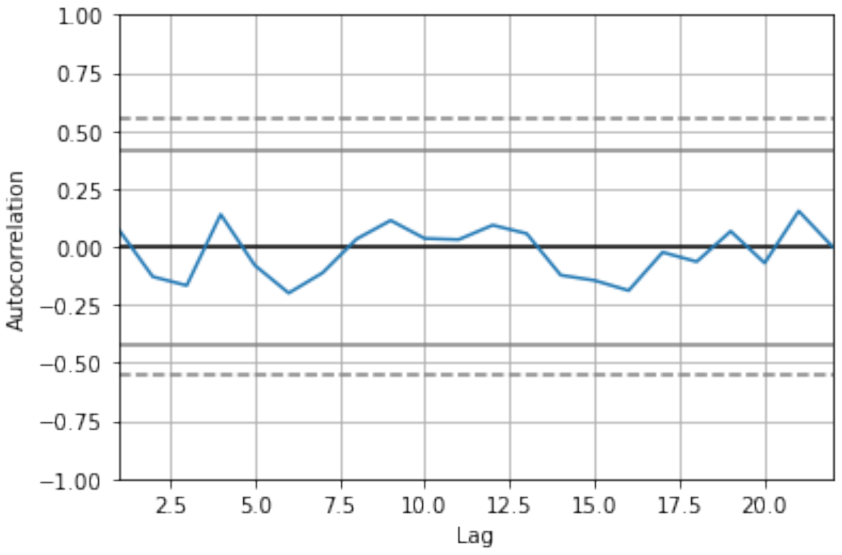
SARIMAX Results

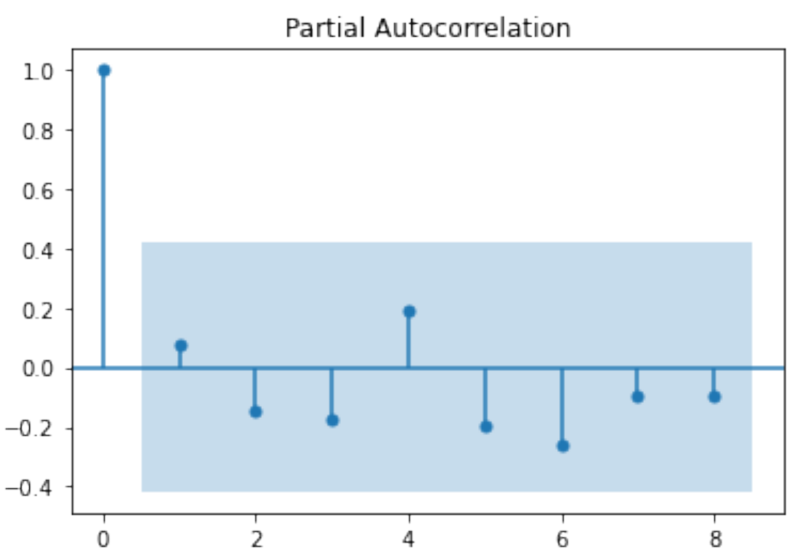
Dep. Variable:	Adj Close**	**No. Observations:**	22
Model:	ARIMA(0, 0, 1)	**Log Likelihood**	72.698
Date:	Tue, 12 Jan 2021	**AIC**	-139.397
Time:	14:12:02	**BIC**	-136.124
Sample:	10-02-2019	**HQIC**	-138.626
	- 10-31-2019		
Covariance Type:	opg		

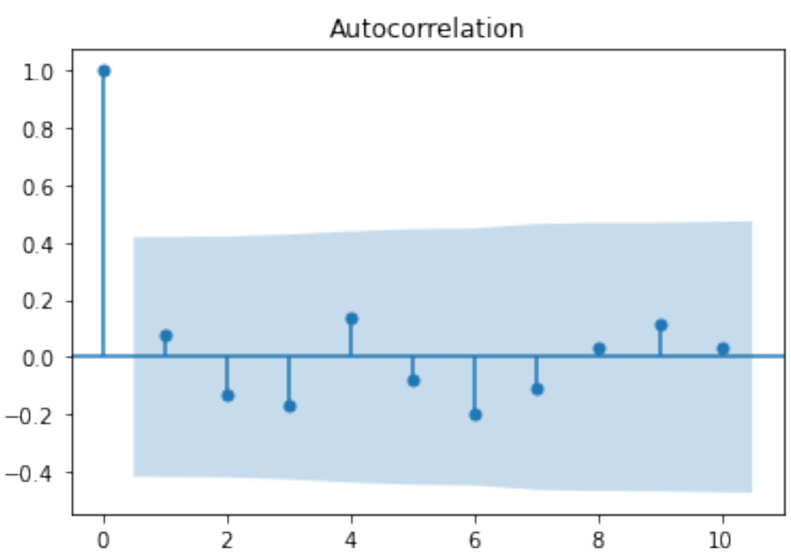
	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.0009	0.004	-0.211	0.833	-0.010	0.008
ma.L1	0.2267	0.522	0.434	0.664	-0.796	1.250
sigma2	7.874e-05	3.92e-05	2.010	0.044	1.96e-06	0.000

Ljung-Box (L1) (Q):	0.35	**Jarque-Bera (JB):**	30.94
Prob(Q):	0.56	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.31	**Skew:**	-1.93
Prob(H) (two-sided):	0.14	**Kurtosis:**	7.35

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



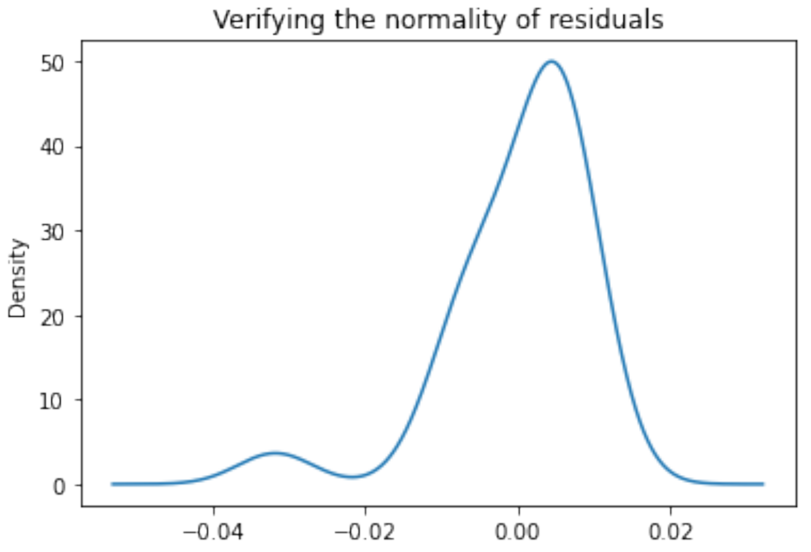




In [155]:

```
print('Standard deviation of the residuals is :',arimaoctequityreturnfit.resid.std())
arimaoctequityreturnfit.resid.plot(kind = 'kde', title = 'Verifying the normality of residuals');
```

```
Standard deviation of the residuals is : 0.009208070945655369
```



In [156]:
```
#Measuring the Equity ETF ARIMA stationarity
adfuller(df_equityetf_daily_return['Nov-2019'])
#We notice that the time series is stationary and the p-value is much less than the critical value which indicates that we reject the null hypothesis and
```

Out[156]:
```
(0.27851102411347545,
 0.9763368175656957,
 8,
 12,
 {'1%': -4.137829282407408,
  '5%': -3.1549724074074077,
  '10%': -2.7144769444444443},
 -103.1558591032435)
```

In [157]:
```
adfuller(df_equityetf_daily_return['Nov-2019'].diff().dropna())
```

Out[157]:
```
(-3.058282465856081,
 0.02979850494060524,
 7,
 12,
 {'1%': -4.137829282407408,
  '5%': -3.1549724074074077,
  '10%': -2.7144769444444443},
 -95.39699637814)
```

In [158]:
```
autocorrelation_plot(df_equityetf_daily_return['Nov-2019'], label = 'Auto-correlation for Nov Equity ETF')
#autocorrelation_plot(df_equityetf_daily_return['Nov-2019'].diff().dropna(), label = 'Auto-correlation for Nov Equity ETF - 1st order differencing')
plot_pacf(df_equityetf_daily_return['Nov-2019'],lags = 8);
plot_acf(df_equityetf_daily_return['Nov-2019'], lags = 10);
arimanovequityreturn = ARIMA(df_equityetf_daily_return['Nov-2019'], order = (0,0,1))
arimanovequityreturnfit = arimanovequityreturn.fit()
arimanovequityreturnfit.summary()
```

Out[158]:

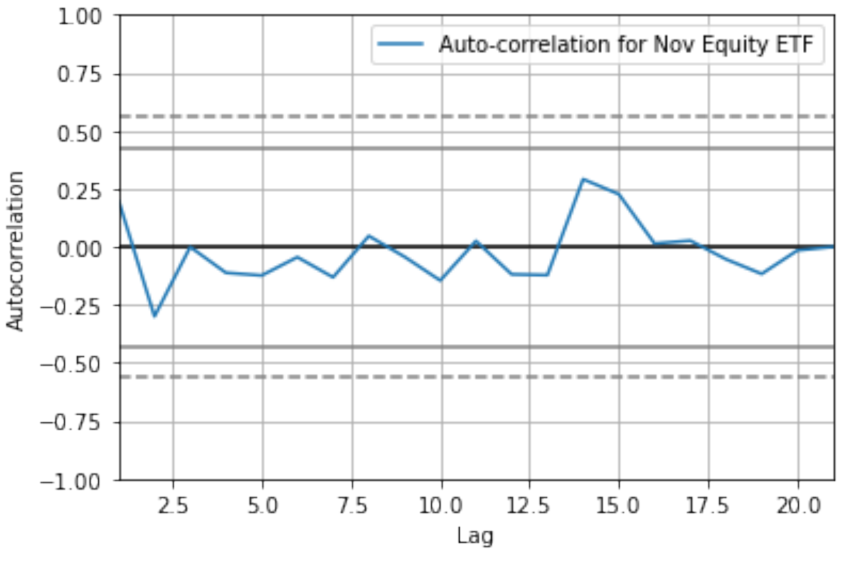
SARIMAX Results

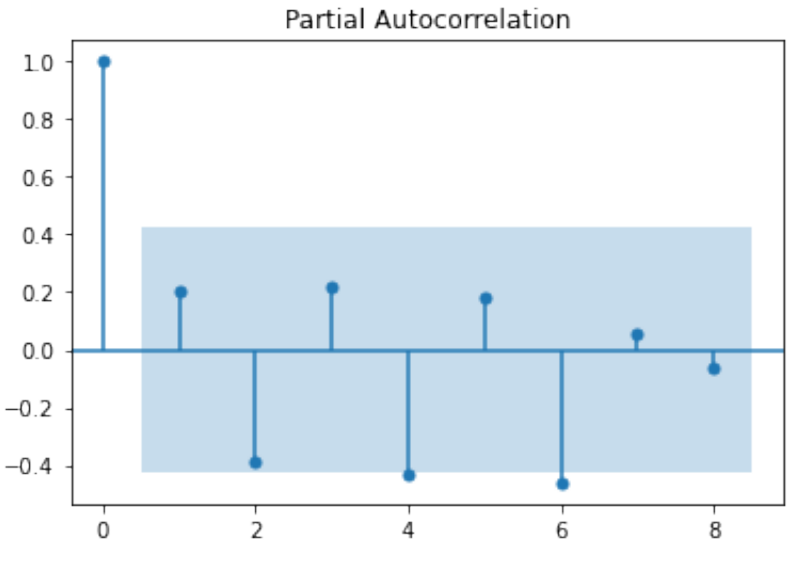
Dep. Variable:	Adj Close**	**No. Observations:**	21
Model:	ARIMA(0, 0, 1)	**Log Likelihood**	81.650
Date:	Tue, 12 Jan 2021	**AIC**	-157.300
Time:	14:12:03	**BIC**	-154.166
Sample:	11-01-2019	**HQIC**	-156.620
	- 11-29-2019		
Covariance Type:	opg		

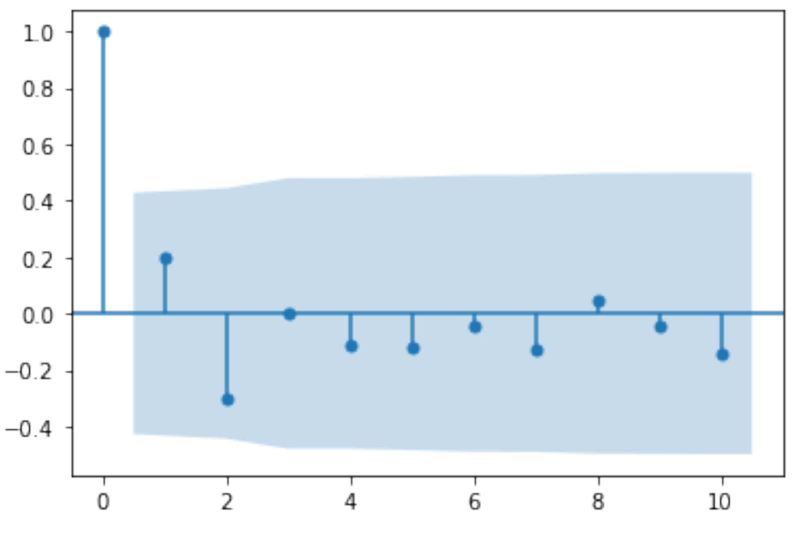
	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0004	0.002	0.207	0.836	-0.003	0.004
ma.L1	0.8055	0.289	2.790	0.005	0.240	1.371
sigma2	2.334e-05	7.35e-06	3.174	0.002	8.93e-06	3.77e-05

Ljung-Box (L1) (Q):	0.64	**Jarque-Bera (JB):**	0.35
Prob(Q):	0.42	**Prob(JB):**	0.84
Heteroskedasticity (H):	1.43	**Skew:**	0.29
Prob(H) (two-sided):	0.65	**Kurtosis:**	3.25

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



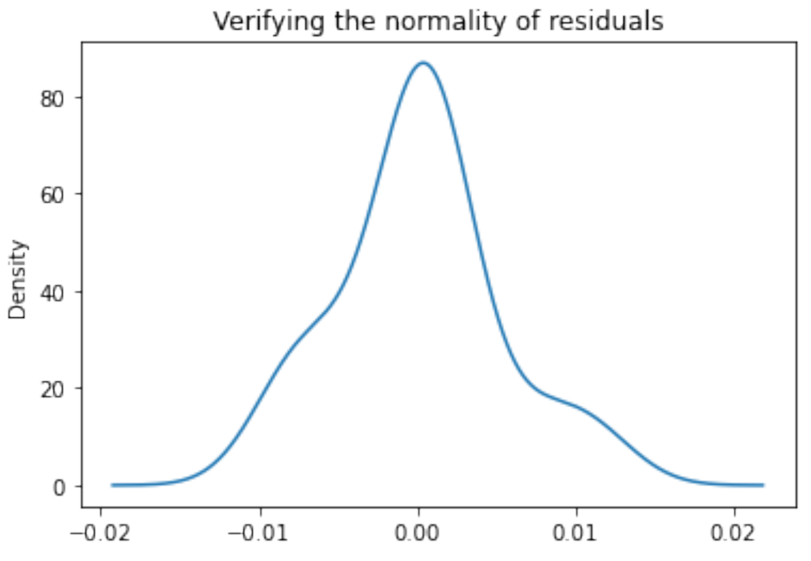




In [159]:

```
print('Standard deviation of the residuals is :',arimanovequityreturnfit.resid.std())
arimanovequityreturnfit.resid.plot(kind = 'kde', title = 'Verifying the normality of residuals');
```

```
Standard deviation of the residuals is : 0.005062396645891091
```



In [160]:
```
df_dailyhighlowgoldetf = df_goldetf['High']-df_goldetf['Low']
'''Using gold ETF prices, find the daily high minus low for each month.
Compute the average for October.
Compute the average for November.'''
df_dailyhighlowgoldetf['Oct-2019']
```

Out[160]:
```
Date
2019-10-01    2.39
2019-10-02    1.42
2019-10-03    1.96
2019-10-04    1.17
2019-10-07    1.53
2019-10-08    1.12
2019-10-09    0.96
2019-10-10    1.08
2019-10-11    1.28
2019-10-14    0.48
2019-10-15    1.09
2019-10-16    0.83
2019-10-17    0.70
2019-10-18    0.39
2019-10-21    1.16
2019-10-22    0.58
2019-10-23    0.50
2019-10-24    0.81
2019-10-25    1.43
2019-10-28    0.85
2019-10-29    0.85
2019-10-30    1.49
2019-10-31    0.80
dtype: float64
```

In [161]:
```
df_dailyhighlowgoldetf['Nov-2019']
```

Out[161]:
```
Date
2019-11-01    0.89
2019-11-04    0.71
2019-11-05    1.51
2019-11-06    0.79
2019-11-07    2.60
2019-11-08    0.94
2019-11-11    1.35
2019-11-12    1.40
2019-11-13    0.58
2019-11-14    1.07
2019-11-15    0.45
2019-11-18    0.55
2019-11-19    0.84
2019-11-20    0.90
2019-11-21    0.79
2019-11-22    0.80
2019-11-25    0.63
2019-11-26    1.22
2019-11-27    0.38
2019-11-29    1.05
dtype: float64
```

In [162]:
```
#Using the gold ETF returns, find the standard deviation for October.  Repeat for November
df_goldetfreturnsstddev = df_goldetf_daily_return.resample('M').std()
df_goldetfreturnsstddev.index = ['October', 'November']
df_goldetfreturnsstddev
```

Out[162]:
```
October     0.006170
November    0.006217
Name: Adj Close**, dtype: float64
```

In [163]:
```
'''Using equity ETF prices, find the daily high minus low for each month.
Compute the average for October.  Compute the average for November.'''
df_dailyhighlowequityetf = df_equityetf['High']-df_equityetf['Low']
```

In [164]:
```
df_dailyhighlowequityetf['Oct-2019']
```

Out[164]:
```
Date
2019-10-01    110.00
2019-10-02     76.00
2019-10-03    114.00
2019-10-04     24.00
2019-10-07     72.08
2019-10-08      0.00
2019-10-09     58.00
2019-10-10     74.00
2019-10-11    110.00
2019-10-14     12.00
2019-10-15     96.00
2019-10-16     70.00
2019-10-17     20.00
2019-10-18     44.00
2019-10-21     79.38
2019-10-22     62.00
2019-10-23      0.00
2019-10-24     30.00
2019-10-25     46.00
2019-10-28    100.00
2019-10-29     22.00
2019-10-30     38.00
2019-10-31     40.00
dtype: float64
```

In [165]:
```
df_dailyhighlowequityetf['Nov-2019']
```

Out[165]:
```
Date
2019-11-01     24.00
2019-11-04    114.00
2019-11-05     34.00
2019-11-06     16.00
2019-11-07     44.00
2019-11-08     26.00
2019-11-11     40.00
2019-11-12     26.00
2019-11-13      8.00
2019-11-14     58.00
2019-11-15     97.90
2019-11-18     28.00
2019-11-19     82.00
2019-11-20     68.04
2019-11-21     34.00
2019-11-22     34.00
2019-11-25     18.00
2019-11-26     42.00
2019-11-27     12.00
2019-11-28      0.00
2019-11-29     72.00
dtype: float64
```

In [166]:
```
#Average of difference between High-Low for each month
df_highlowgoldequitymean = df_dailyhighlowequityetf.resample('M').mean()
df_highlowgoldequitymean.index = ['October', 'November']
df_highlowgoldequitymean
```

Out[166]:
```
October     56.411304
November    41.806667
dtype: float64
```

In [167]:
```
#Using equity ETF returns, find the standard deviation for October.  Repeat for November
df_equityetfreturnsstddev = df_equityetf_daily_return.resample('M').std()
df_equityetfreturnsstddev.index = ['October', 'November']
df_equityetfreturnsstddev
```

Out[167]:
```
October     0.009191
November    0.005756
Name: Adj Close**, dtype: float64
```

In [168]:
```
#GARCH(1,1) Model on the Residuals of the ARIMA(1,0,0) model - October 2019 Gold ETF
print("GARCH(1,1) Model on the Residuals of the ARIMA(1,0,0) model - October 2019 Gold ETF")
print(" ")
garch_goldetf_oct = arch.arch_model(arimaoctgoldetfreturnfit.resid, p=1, q=1)
garch_goldetf_oct_fit = garch_goldetf_oct.fit()
print(garch_goldetf_oct_fit.summary())
```

```
GARCH(1,1) Model on the Residuals of the ARIMA(1,0,0) model - October 2019 Gold ETF

Iteration:      1,   Func. Count:      6,   Neg. LLF: 16706906.132050708
Iteration:      2,   Func. Count:     16,   Neg. LLF: -81.27372852529649
Optimization terminated successfully    (Exit mode 0)
            Current function value: -81.27372855690483
            Iterations: 6
            Function evaluations: 16
            Gradient evaluations: 2
                     Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:                   None   R-squared:                      -0.000
Mean Model:             Constant Mean   Adj. R-squared:                 -0.000
Vol Model:                      GARCH   Log-Likelihood:                81.2737
Distribution:                  Normal   AIC:                          -154.547
Method:            Maximum Likelihood   BIC:                          -150.183
                                        No. Observations:                   22
Date:                Tue, Jan 12 2021   Df Residuals:                       18
Time:                        14:12:05   Df Model:                            4
                                 Mean Model
==============================================================================
                  coef    std err          t      P>|t|       95.0% Conf. Int.
------------------------------------------------------------------------------
mu         -4.5935e-05  1.487e-03 -3.089e-02      0.975 [-2.961e-03,2.869e-03]
                              Volatility Model
============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
omega      1.0896e-05  3.878e-10  2.809e+04      0.000 [1.090e-05,1.090e-05]
alpha[1]   1.0000e-02  8.485e-02      0.118      0.906     [ -0.156,  0.176]
beta[1]        0.6900  9.028e-02      7.643  2.121e-14     [  0.513,  0.867]
============================================================================

Covariance estimator: robust
```

In [169]:
```
#GARCH(1,1) Model on the Residuals of the ARIMA(1,1,0) model - November 2019 Gold ETF
print("GARCH(1,1) Model on the Residuals of the ARIMA(1,1,0) model - November 2019 Gold ETF")
print(" ")
garch_goldetf_nov = arch.arch_model(arimanovgoldetfreturnfit.resid, p=1, q=1)
garch_goldetf_nov_fit = garch_goldetf_nov.fit()
print(garch_goldetf_nov_fit.summary())
```

```
GARCH(1,1) Model on the Residuals of the ARIMA(1,1,0) model - November 2019 Gold ETF

Iteration:      1,   Func. Count:      6,   Neg. LLF: 551198191965.6055
Iteration:      2,   Func. Count:     16,   Neg. LLF: -72.86969399322018
Optimization terminated successfully    (Exit mode 0)
            Current function value: -72.86969402910499
            Iterations: 6
            Function evaluations: 16
            Gradient evaluations: 2
                     Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:                   None   R-squared:                      -0.000
Mean Model:             Constant Mean   Adj. R-squared:                 -0.000
Vol Model:                      GARCH   Log-Likelihood:                72.8697
Distribution:                  Normal   AIC:                          -137.739
Method:            Maximum Likelihood   BIC:                          -133.756
                                        No. Observations:                   20
Date:                Tue, Jan 12 2021   Df Residuals:                       16
Time:                        14:12:05   Df Model:                            4
                                  Mean Model
=============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
-----------------------------------------------------------------------------
mu         7.5171e-05  1.307e-03  5.752e-02      0.954 [-2.486e-03,2.636e-03]
                              Volatility Model
============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
omega      1.2301e-05  1.552e-10  7.926e+04      0.000 [1.230e-05,1.230e-05]
alpha[1]       0.0500      0.104      0.481      0.630     [ -0.154,  0.254]
beta[1]        0.6500      0.150      4.340  1.428e-05     [  0.356,  0.944]
============================================================================

Covariance estimator: robust
```

In [170]:

```python
#GARCH(1,1) Model on the Residuals of the ARIMA(0,0,1) model - October 2019 Equity ETF
print("GARCH(1,1) Model on the Residuals of the ARIMA(0,0,1) model - October 2019 Equity ETF")
print(" ")
garch_equityetf_oct = arch.arch_model(arimaoctequityreturnfit.resid, p=1, q=1)
garch_equityetf_oct_fit = garch_equityetf_oct.fit()
print(garch_equityetf_oct_fit.summary())
```

```
GARCH(1,1) Model on the Residuals of the ARIMA(0,0,1) model - October 2019 Equity ETF

Iteration:      1,   Func. Count:      6,   Neg. LLF: 1308492235611928.8
Iteration:      2,   Func. Count:     16,   Neg. LLF: -74.01514801493096
Optimization terminated successfully    (Exit mode 0)
            Current function value: -74.01514811543122
            Iterations: 6
            Function evaluations: 16
            Gradient evaluations: 2
                     Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:                   None   R-squared:                      -0.001
Mean Model:             Constant Mean   Adj. R-squared:                 -0.001
Vol Model:                      GARCH   Log-Likelihood:                74.0151
Distribution:                  Normal   AIC:                          -140.030
Method:            Maximum Likelihood   BIC:                          -135.666
                                        No. Observations:                   22
Date:                Tue, Jan 12 2021   Df Residuals:                       18
Time:                        14:12:05   Df Model:                            4
                                  Mean Model
=============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
-----------------------------------------------------------------------------
mu         5.2158e-04  1.389e-03      0.375      0.707 [-2.201e-03,3.244e-03]
                              Volatility Model
============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
omega      2.4280e-05  1.433e-06     16.945  2.085e-64 [2.147e-05,2.709e-05]
alpha[1]       0.2000      0.236      0.846      0.397     [ -0.263,  0.663]
beta[1]        0.5000      0.212      2.354  1.857e-02   [8.370e-02,  0.916]
============================================================================

Covariance estimator: robust
```

In [171]:

```python
#GARCH(1,1) Model on the Residuals of the ARIMA(0,0,1) model - November 2019 Equity ETF
print("GARCH(1,1) Model on the Residuals of the ARIMA(0,0,1) model - November 2019 Equity ETF")
print(" ")

garch_equityetf_nov = arch.arch_model(arimanovequityreturnfit.resid, p=1, q=1)
garch_equityetf_nov_fit = garch_equityetf_nov.fit()
print(garch_equityetf_nov_fit.summary())
```

```
GARCH(1,1) Model on the Residuals of the ARIMA(0,0,1) model - November 2019 Equity ETF

Iteration:      1,   Func. Count:      6,   Neg. LLF: 6054389.094355717
Optimization terminated successfully    (Exit mode 0)
            Current function value: -81.64083316845387
            Iterations: 1
            Function evaluations: 12
            Gradient evaluations: 1
                     Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:                   None   R-squared:                      -0.000
Mean Model:             Constant Mean   Adj. R-squared:                 -0.000
Vol Model:                      GARCH   Log-Likelihood:                81.6408
Distribution:                  Normal   AIC:                          -155.282
Method:            Maximum Likelihood   BIC:                          -151.104
                                        No. Observations:                   21
Date:                Tue, Jan 12 2021   Df Residuals:                       17
Time:                        14:12:05   Df Model:                            4
                                  Mean Model
=============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
-----------------------------------------------------------------------------
mu         1.2329e-04  1.119e-03      0.110      0.912 [-2.069e-03,2.316e-03]
                               Volatility Model
=============================================================================
                 coef    std err          t      P>|t|       95.0% Conf. Int.
-----------------------------------------------------------------------------
omega      1.2204e-05  1.038e-09  1.176e+04      0.000  [1.220e-05,1.221e-05]
alpha[1]   1.0000e-02  1.401e-02      0.714      0.475 [-1.745e-02,3.745e-02]
beta[1]        0.4900      0.178      2.750  5.965e-03      [  0.141,  0.839]
=============================================================================

Covariance estimator: robust
```

In [172]:

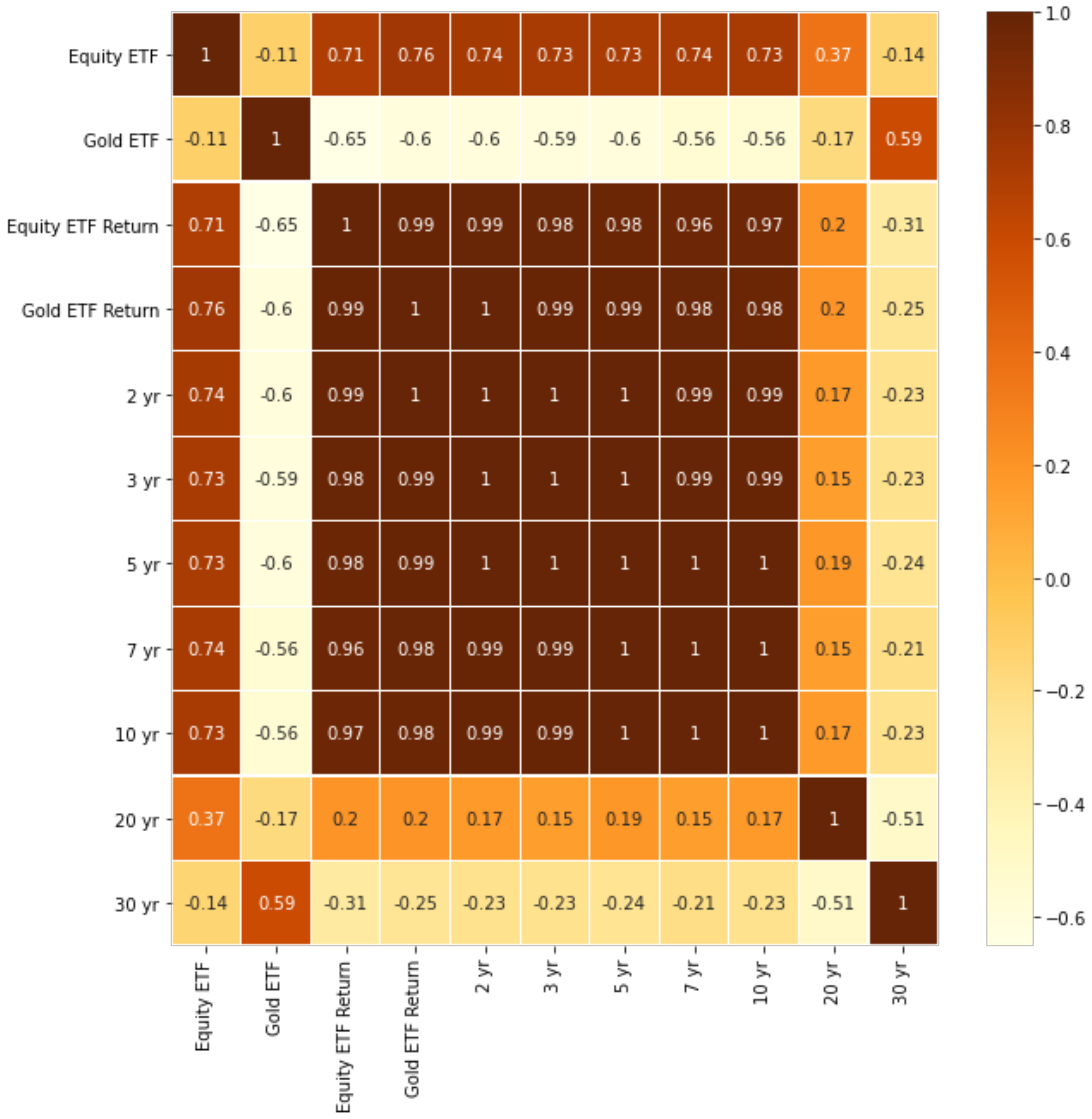
```python
print('The pearsons correlation coefficient between Gold and Equity ETF for the month of Oct-2019 is:',np.corrcoef(df_goldetf_daily_return['Oct-2019'], c
```

```
The pearsons correlation coefficient between Gold and Equity ETF for the month of Oct-2019 is: -0.5107430668448303
```

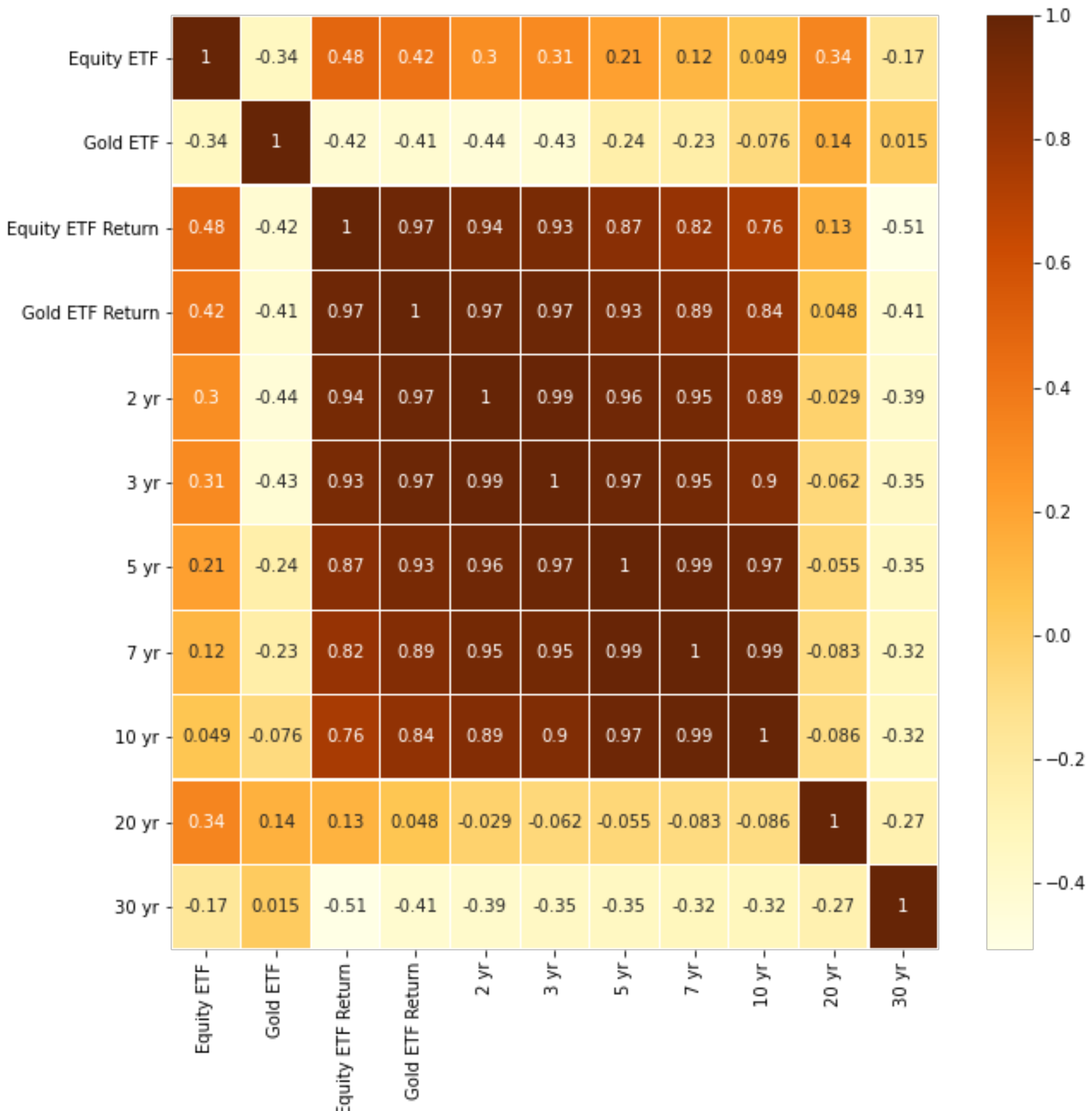
In [173]:
```
print('The pearsons correlation coefficient between Gold and Equity ETF for the month of Nov-2019 is:',pd.DataFrame(pd.concat([df_equityetf_daily_return[
```

```
The pearsons correlation coefficient between Gold and Equity ETF for the month of Nov-2019 is: -0.25150272919698863
```

In [174]:
```
df_all = pd.concat([df_equityetf['Adj Close**'], df_goldetf['Adj Close**'], df_trate/100, df_equityetf_daily_return, df_goldetf_daily_return],join = 'inn
df_all.columns = ['Equity ETF','Gold ETF','Equity ETF Return','Gold ETF Return','2 yr','3 yr', '5 yr','7 yr', '10 yr','20 yr','30 yr']
df_all['Oct-2019'].corr(method = 'pearson')

import seaborn as sns
fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(df_all['Oct-2019'].corr(method = 'pearson'), linewidths=0.1, cmap = 'YlOrBr', annot=True);
```

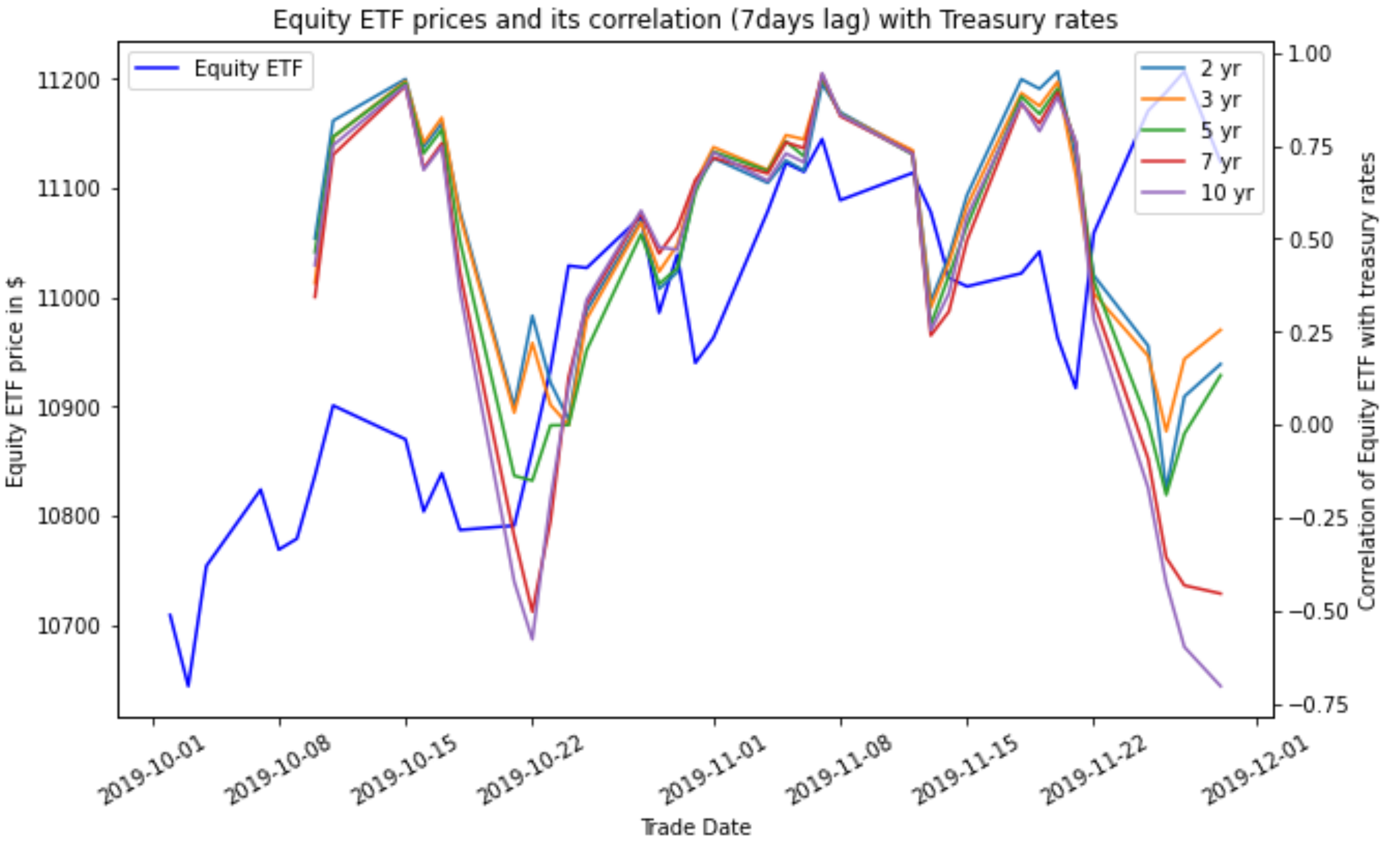


In [175]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(df_all['Nov-2019'].corr(method = 'pearson'), linewidths=0.1, cmap = 'YlOrBr', annot=True);
```



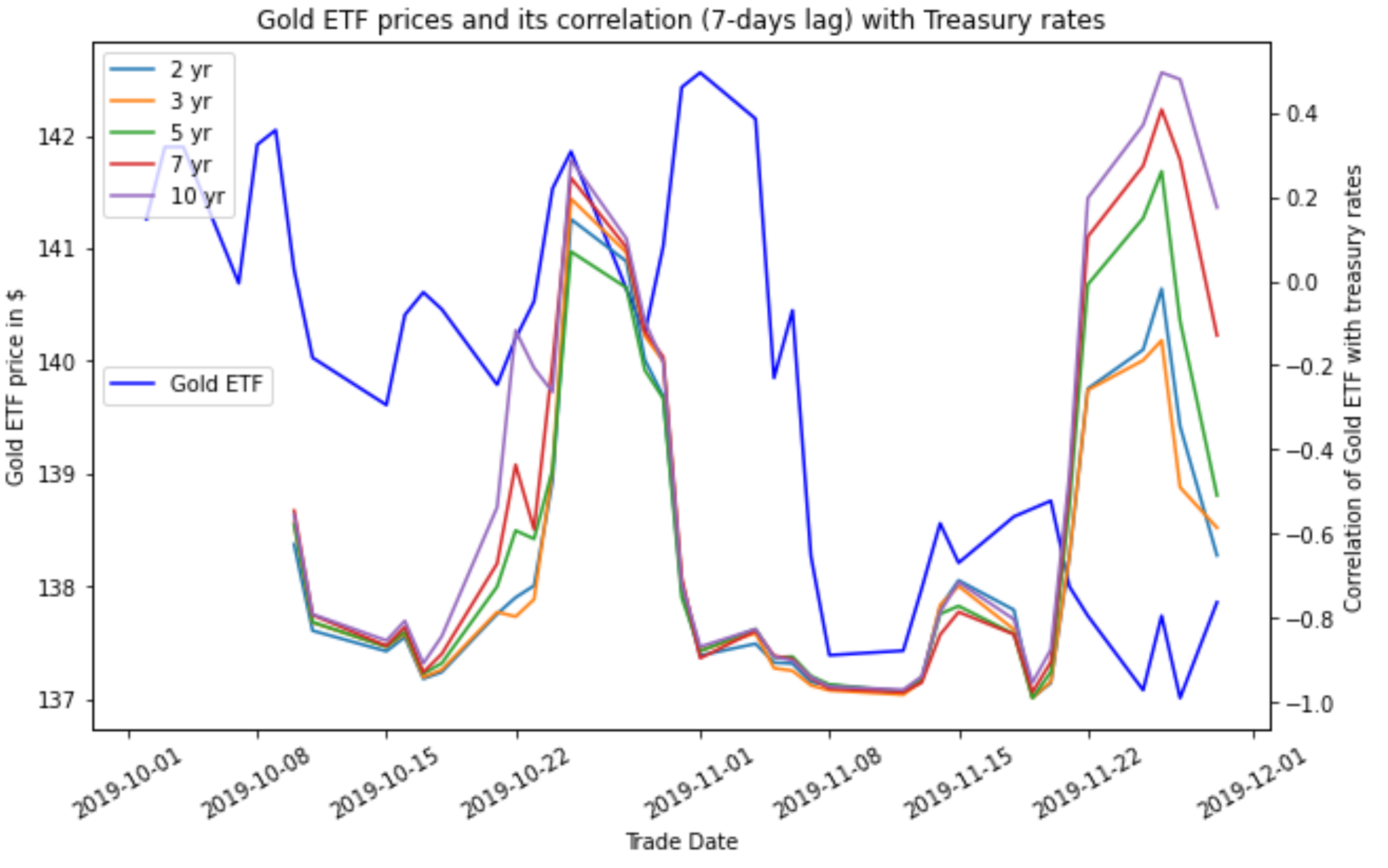
In [176]:
```
#Equally weighted lag - 7 days correlation between Equity ETF with yields
corr_equity_etf = df_all[['Equity ETF','2 yr','3 yr', '5 yr','7 yr', '10 yr']].rolling(7).corr().dropna().xs('Equity ETF', level=1).iloc[:,1:]
corr_equity_etf = pd.DataFrame(corr_equity_etf)
```

In [177]:
```
#Plotting the gold and equity ETF
mondays = WeekdayLocator(MONDAY)
fig, ax1 = plt.subplots(figsize=(10,6))
ax1.set_title("Equity ETF prices and its correlation (7days lag) with Treasury rates")
lns1 = ax1.plot(df_all['Equity ETF'], color = 'Blue', label = "Equity ETF")
ax1.set_xlabel("Trade Date")
plt.xticks(rotation=30)
ax1.xaxis.set_major_locator(mondays)
ax1.set_ylabel("Equity ETF price in $")
ax2 = ax1.twinx()
ax2.set_ylabel("Correlation of Equity ETF with treasury rates")

labels=['2 yr','3 yr', '5 yr','7 yr', '10 yr']
i=0
for label in labels:
    ax2.plot(corr_equity_etf.iloc[:,i], label=label)
    i+=1
ax1.legend()
ax2.legend()
plt.show()
```



In [178]:
```
#Equally weighted lag - 7 dayscorrelation between Equity ETF with yields
corr_gold_etf = df_all[['Gold ETF','2 yr','3 yr', '5 yr','7 yr', '10 yr']].rolling(7).corr().dropna().xs('Gold ETF', level=1).iloc[:,1:]
corr_gold_etf = pd.DataFrame(corr_gold_etf)
```

In [182]:
```
#Plotting the gold and equity ETF
mondays = WeekdayLocator(MONDAY)
fig, ax1 = plt.subplots(figsize=(10,6))
ax1.set_title("Gold ETF prices and its correlation (7-days lag) with Treasury rates")
lns1 = ax1.plot(df_all['Gold ETF'], color = 'Blue', label = "Gold ETF")
ax1.set_xlabel("Trade Date")
plt.xticks(rotation=30)
ax1.xaxis.set_major_locator(mondays)
ax1.set_ylabel("Gold ETF price in $")
ax2 = ax1.twinx()
ax2.set_ylabel("Correlation of Gold ETF with treasury rates")

labels=['2 yr','3 yr', '5 yr','7 yr', '10 yr']
i=0
for label in labels:
    ax2.plot(corr_gold_etf.iloc[:,i], label=label)
    i+=1
ax1.legend(loc = 6)
ax2.legend()
plt.show()
```



In [ ]: