A comparitive analysis of some related Finnish Companies in the Helsinki Stock Exchange("UPM.HE,STERV.HE,METSO.HE,,FORTUM.HE).UPM and STERV were highly correlated to each other in this project.

In [1]:

```
import numpy as np
import pandas as pd
import scipy
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
import statsmodels.graphics.tsaplots as sgt
import statsmodels.tsa.stattools as sts
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from pmdarima.arima import auto_arima
from arch import arch_model
import yfinance
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
df = yfinance.download (tickers = "UPM.HE,STERV.HE,METSO.HE,,FORTUM.HE", start = "2000-01-07",
                               end = "2020-09-18", interval = "1d", group_by = 'ticker', auto_adjus
= True, treads = True)
```

```
[*********************100%***********************]  4 of 4 completed
```

In [3]:

```
#We will be concentrating on the Closing prices
df['UPM_CL'] = df['UPM.HE'].Close[:]
df['FORTUM_CL'] = df['FORTUM.HE'].Close[:]
df['STERV_CL'] = df['STERV.HE'].Close[:]
df['METSO_CL'] = df['METSO.HE'].Close[:]
```

In [4]:

```
df = df.iloc[1:]
del df['UPM.HE']
del df['FORTUM.HE']
del df['STERV.HE']
del df['METSO.HE']
df=df.asfreq('b')
df=df.fillna(method='ffill')
```

In [5]:

```
df.head()
```

Out[5]:

	UPM_CL	FORTUM_CL	STERV_CL	METSO_CL
Date				
2000-01-10	6.582538	0.930695	7.386399	3.522561
2000-01-11	6.304402	0.915087	7.282366	3.407404
2000-01-12	5.995362	0.911183	7.078460	3.477912
2000-01-13	6.179241	0.926794	7.032685	3.560141

	UPM_CL	FORTUM_CL	STERV_CL	METSO_CL
2000-01-13	6.179241	0.926794	7.032685	3.560141
2000-01-14	6.335306	0.926794	6.907845	3.524886

RETURNS

In [6]:

```
df['ret_UPM'] = df.UPM_CL.pct_change(1).mul(100)
df['ret_FORTUM'] = df.FORTUM_CL.pct_change(1).mul(100)
df['ret_STERV'] = df.STERV_CL.pct_change(1).mul(100)
df['ret_METSO'] = df.METSO_CL.pct_change(1).mul(100)
```

In [7]:

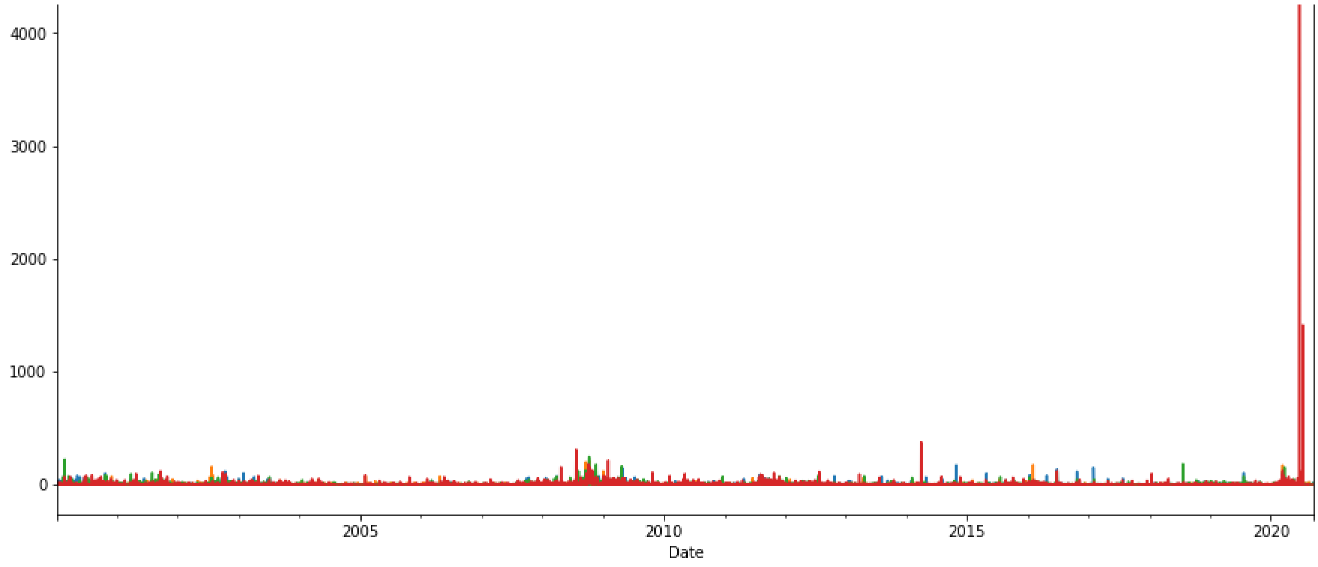
```
# Creating Squared Returns
df['sq_UPM'] = (df.ret_UPM)**2
df['sq_FORTUM'] = (df.ret_FORTUM)**2
df['sq_STERV'] = (df.ret_STERV)**2
df['sq_METSO'] = (df.ret_METSO)**2
```

In [8]:

```
#plotting closing prices
df['UPM_CL'].plot(legend=True,figsize=(15,8))
df['FORTUM_CL'].plot(legend=True)
df['STERV_CL'].plot(legend=True)
df['METSO_CL'].plot(legend=True);
```

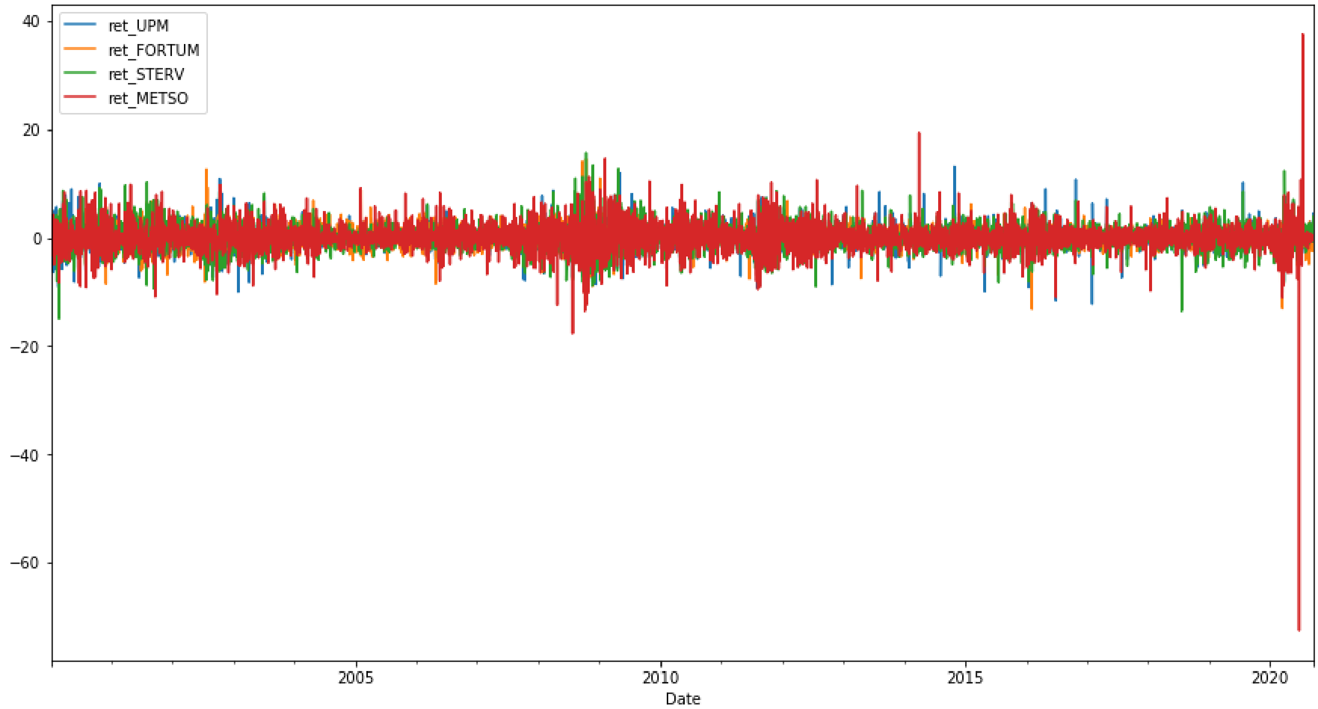
In [9]:

```
#plotting sqaure returns(volatility)
df['sq_UPM'].plot(legend=True,figsize=(15,8))
df['sq_FORTUM'].plot(legend=True)
df['sq_STERV'].plot(legend=True)
df['sq_METSO'].plot(legend=True);
```



In [10]:

```
#plotting returns
df['ret_UPM'].plot(legend=True,figsize=(15,8))
df['ret_FORTUM'].plot(legend=True)
df['ret_STERV'].plot(legend=True)
df['ret_METSO'].plot(legend=True);
```



In [11]:

```
df_co=df[['UPM_CL','FORTUM_CL','STERV_CL','METSO_CL']].corr()
df_co
```

Out[11]:

	UPM_CL	FORTUM_CL	STERV_CL	METSO_CL
UPM_CL	1.000000	0.749284	0.932792	0.778270
FORTUM_CL	0.749284	1.000000	0.599455	0.891129
STERV_CL	0.932792	0.599455	1.000000	0.628190
METSO_CL	0.778270	0.891129	0.628190	1.000000

In [12]:

```
df_rt=df[['ret_UPM','ret_FORTUM','ret_STERV','ret_METSO']].corr()
df_rt
```

Out[12]:

	ret_UPM	ret_FORTUM	ret_STERV	ret_METSO
ret_UPM	1.000000	0.332242	0.794739	0.406256
ret_FORTUM	0.332242	1.000000	0.346507	0.341670
ret_STERV	0.794739	0.346507	1.000000	0.428692
ret_METSO	0.406256	0.341670	0.428692	1.000000

In [13]:

```
df_vol=df[['sq_UPM','sq_FORTUM','sq_STERV','sq_METSO']].corr()
df_vol
```

Out[13]:

	sq_UPM	sq_FORTUM	sq_STERV	sq_METSO
sq_UPM	1.000000	0.253043	0.632522	0.051457
sq_FORTUM	0.253043	1.000000	0.358592	0.071432
sq_STERV	0.632522	0.358592	1.000000	0.053771
sq_METSO	0.051457	0.071432	0.053771	1.000000

SPLITTING OUR DATA TO TRAIN AND TEST SET

In [14]:

```
size = int(len(df)*0.8)
train= df.iloc[:size]
test=df.iloc[size:]
```

In [15]:

```
#lets try to fit in the various forecasting models
# AR(1)
model_ar = ARIMA(train.UPM_CL, order = (1,0,0))
results_ar = model_ar.fit()
```

In [16]:

```
train.tail(2)
```

Out[16]:

	UPM_CL	FORTUM_CL	STERV_CL	METSO_CL	ret_UPM	ret_FORTUM	ret_STERV	ret_METSO	sq_UPM	sq_FORTUM	sq_ST
Date											
2016-07-27	15.219547	10.663694	6.760567	19.772427	-1.564199	0.00000	-0.674433	0.079792	2.446720	0.000000	0.45
2016-07-28	15.336302	10.565413	6.727181	19.394157	0.767135	-0.92164	-0.493831	-1.913115	0.588497	0.849421	0.24

In [17]:

```
start= "2016-07-29"
```

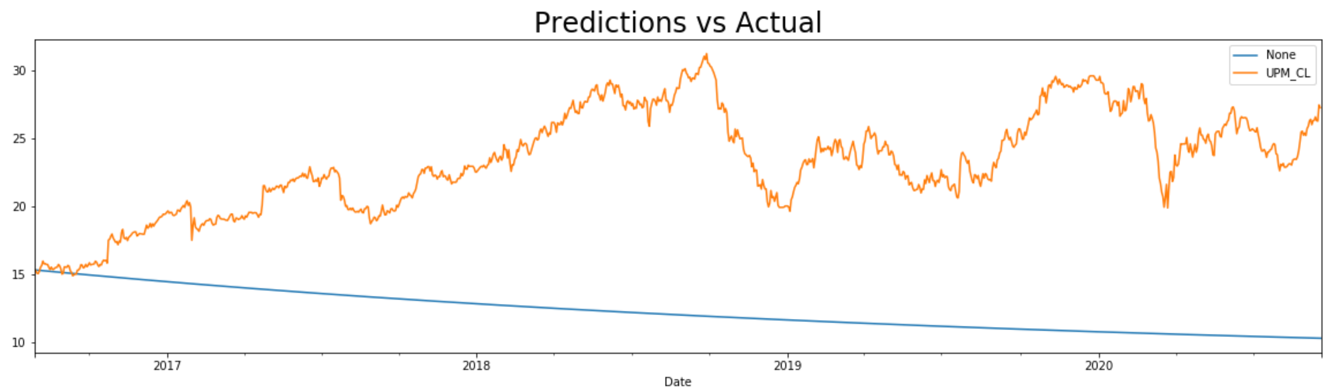
```
end="2017-09-18"
```

In [18]:

```
end='2020-09-18'
df_pred = results_ar.predict(start = start, end = end)
```

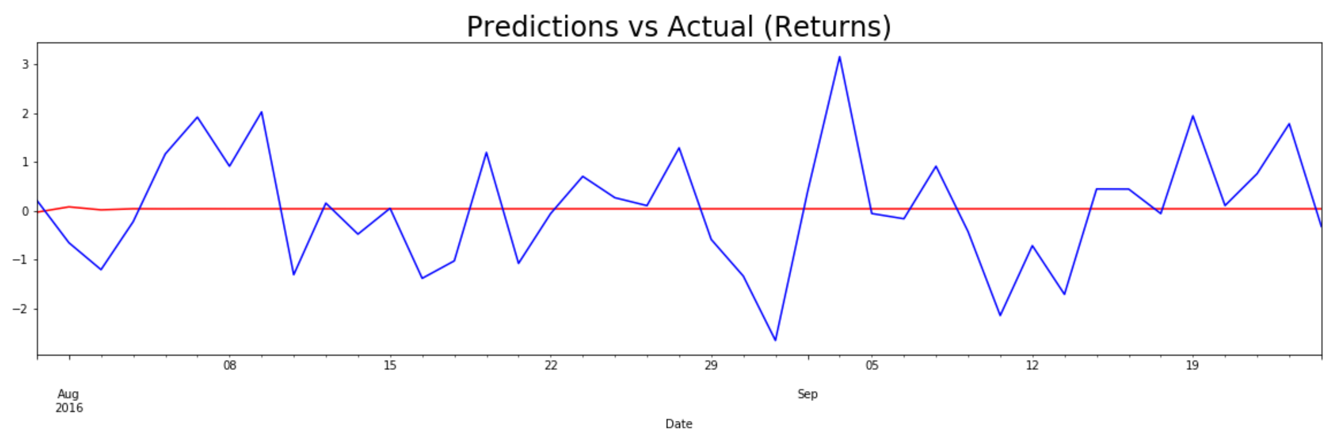
In [19]:

```
df_pred[start:end].plot(figsize = (20,5),legend=True)
test.UPM_CL[start:end].plot(legend=True)
plt.title("Predictions vs Actual", size = 24)
plt.show()
```



In [20]:

```
#using returns AR
end = "2016-09-25"

model_ret_ar = ARIMA(train.ret_UPM[1:], order = (3,0,0))
results_ret_ar = model_ret_ar.fit()

df_pred_ar = results_ret_ar.predict(start = start, end = end)

df_pred_ar[start:end].plot(figsize = (20,5), color = "red")
test.ret_UPM[start:end].plot(color = "blue")
plt.title("Predictions vs Actual (Returns)", size = 24)
plt.show()
```



In [21]:

```
results_ret_ar.summary()
```

Out[21]:

ARMA Model Results

Dep. Variable:	ret_UPM	**No. Observations:**	4318
Model:	ARMA(3, 0)	**Log Likelihood**	-9408.172

Model:	ARMA(3, 0)	**Log Likelihood**	-9408.172
Method:	css-mle	**S.D. of innovations**	2.138
Date:	Wed, 23 Sep 2020	**AIC**	18826.343
Time:	00:33:41	**BIC**	18858.196
Sample:	01-11-2000	**HQIC**	18837.591
	- 07-28-2016		

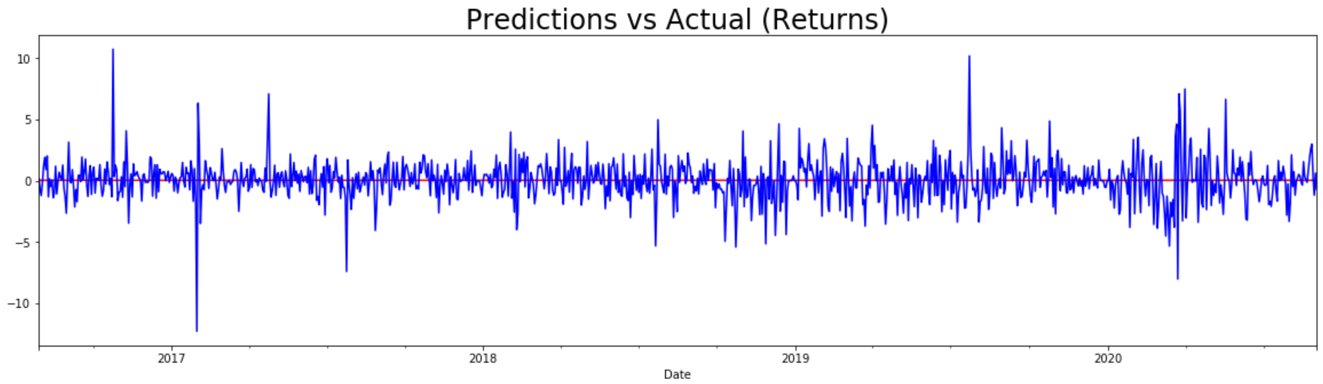
	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0425	0.032	1.324	0.185	-0.020	0.105
ar.L1.ret_UPM	0.0319	0.015	2.097	0.036	0.002	0.062
ar.L2.ret_UPM	-0.0131	0.015	-0.859	0.390	-0.043	0.017
ar.L3.ret_UPM	-0.0321	0.015	-2.110	0.035	-0.062	-0.002

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.4947	-2.6318j	3.0266	-0.1678
AR.2	1.4947	+2.6318j	3.0266	0.1678
AR.3	-3.3964	-0.0000j	3.3964	-0.5000

In [22]:

```
#MA

end= "2020-09-01"

model_ret_ma = ARIMA(train.ret_UPM[1:], order=(0,0,3))
results_ret_ma = model_ret_ma.fit()

df_pred_ma = results_ret_ma.predict(start = start, end = end)

df_pred_ma[start:end].plot(figsize = (20,5), color = "red")
test.ret_UPM[start:end].plot(color = "blue")
plt.title("Predictions vs Actual (Returns)", size = 24)
plt.show()
```



In [23]:

```
results_ret_ma.summary()
```

Out[23]:

ARMA Model Results

Dep. Variable:	ret_UPM	**No. Observations:**	4318
Model:	ARMA(0, 3)	**Log Likelihood**	-9408.091
Method:	css-mle	**S.D. of innovations**	2.138
Date:	Wed, 23 Sep	**AIC**	18826.181

Date:	2020	**AIC**	18826.181
Time:	00:33:42	**BIC**	18858.034
Sample:	01-11-2000	**HQIC**	18837.428
	- 07-28-2016		

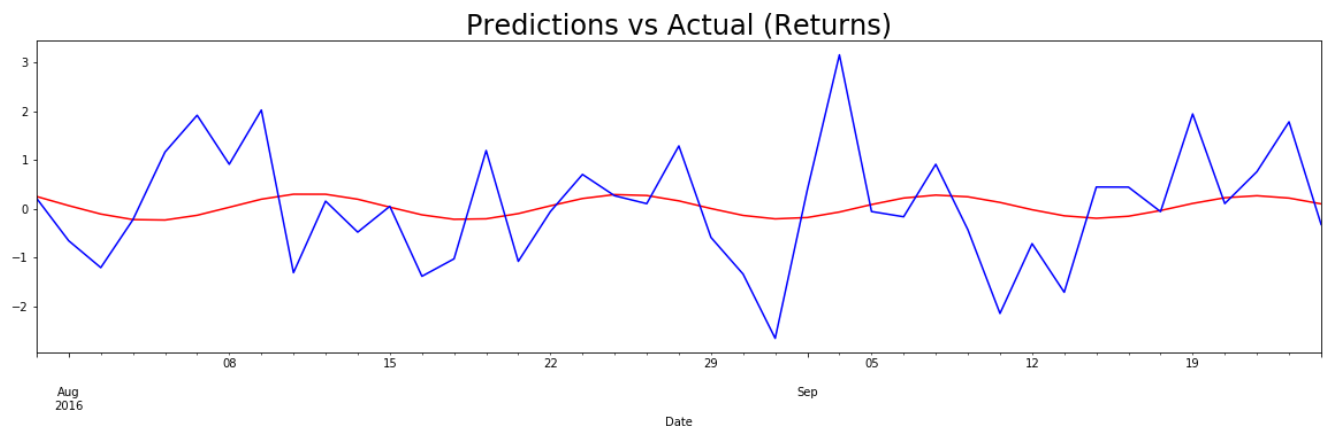
	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0425	0.032	1.326	0.185	-0.020	0.105
ma.L1.ret_UPM	0.0319	0.015	2.097	0.036	0.002	0.062
ma.L2.ret_UPM	-0.0130	0.015	-0.850	0.395	-0.043	0.017
ma.L3.ret_UPM	-0.0337	0.015	-2.204	0.027	-0.064	-0.004

Roots

	Real	Imaginary	Modulus	Frequency
MA.1	3.0694	-0.0000j	3.0694	-0.0000
MA.2	-1.7268	-2.5832j	3.1072	-0.3438
MA.3	-1.7268	+2.5832j	3.1072	0.3438

In [24]:

```
#ARMA
end = "2016-09-25"
model_ret_arma = ARIMA(train.ret_UPM[1:], order=(3,0,3))
results_ret_arma = model_ret_arma.fit()

df_pred_arma = results_ret_arma.predict(start = start, end = end)

df_pred_arma[start:end].plot(figsize = (20,5), color = "red")
test.ret_UPM[start:end].plot(color = "blue")
plt.title("Predictions vs Actual (Returns)", size = 24)
plt.show()
```



In [25]:

```
results_ret_arma.summary()
```

Out[25]:

ARMA Model Results

Dep. Variable:	ret_UPM	**No. Observations:**	4318
Model:	ARMA(3, 3)	**Log Likelihood**	-9403.506
Method:	css-mle	**S.D. of innovations**	2.136
Date:	Wed, 23 Sep 2020	**AIC**	18823.013
Time:	00:33:48	**BIC**	18873.977
Sample:	01-11-2000	**HQIC**	18841.008

- 07-28-2016

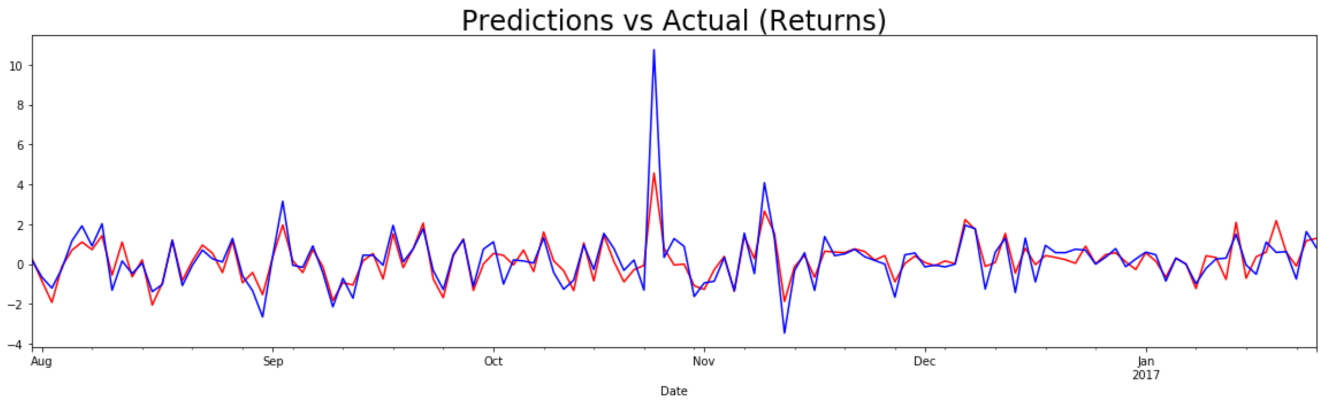
	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0424	0.033	1.278	0.201	-0.023	0.107
ar.L1.ret_UPM	1.5580	0.272	5.720	0.000	1.024	2.092
ar.L2.ret_UPM	-0.9291	0.435	-2.136	0.033	-1.782	-0.077
ar.L3.ret_UPM	-0.0365	0.270	-0.135	0.893	-0.566	0.493
ma.L1.ret_UPM	-1.5284	0.272	-5.628	0.000	-2.061	-0.996
ma.L2.ret_UPM	0.8725	0.435	2.006	0.045	0.020	1.725
ma.L3.ret_UPM	0.0719	0.269	0.267	0.789	-0.456	0.600

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	0.8071	-0.6005j	1.0060	-0.1018
AR.2	0.8071	+0.6005j	1.0060	0.1018
AR.3	-27.0673	-0.0000j	27.0673	-0.5000
MA.1	0.8095	-0.5966j	1.0056	-0.1011
MA.2	0.8095	+0.5966j	1.0056	0.1011
MA.3	-13.7506	-0.0000j	13.7506	-0.5000

In [26]:

```
#ARMA
end = "2017-01-25"
model_ret_armax = ARIMA(train.ret_UPM[1:], exog = train[["ret_FORTUM","ret_METSO","ret_STERV"]][1:]
, order = (1,0,1))
results_ret_armax= model_ret_armax.fit()

df_pred_armax = results_ret_armax.predict(start = start, end = end,
                                          exog =test[["ret_FORTUM","ret_METSO","ret_STERV"]][start:
nd])

df_pred_armax[start:end].plot(figsize = (20,5), color = "red")
test.ret_UPM[start:end].plot(color = "blue")
plt.title("Predictions vs Actual (Returns)", size = 24)
plt.show()
```



In [27]:

```
results_ret_armax.summary()
```

Out[27]:

ARMA Model Results

Dep. Variable:	ret_UPM	**No. Observations:**	4318
Model:	ARMA(1, 1)	**Log Likelihood**	-7194.434
Method:	css-mle	**S.D. of innovations**	1.280

		innovations	
Date:	Wed, 23 Sep 2020	**AIC**	14402.869
Time:	00:33:50	**BIC**	14447.462
Sample:	01-11-2000	**HQIC**	14418.615
	- 07-28-2016		

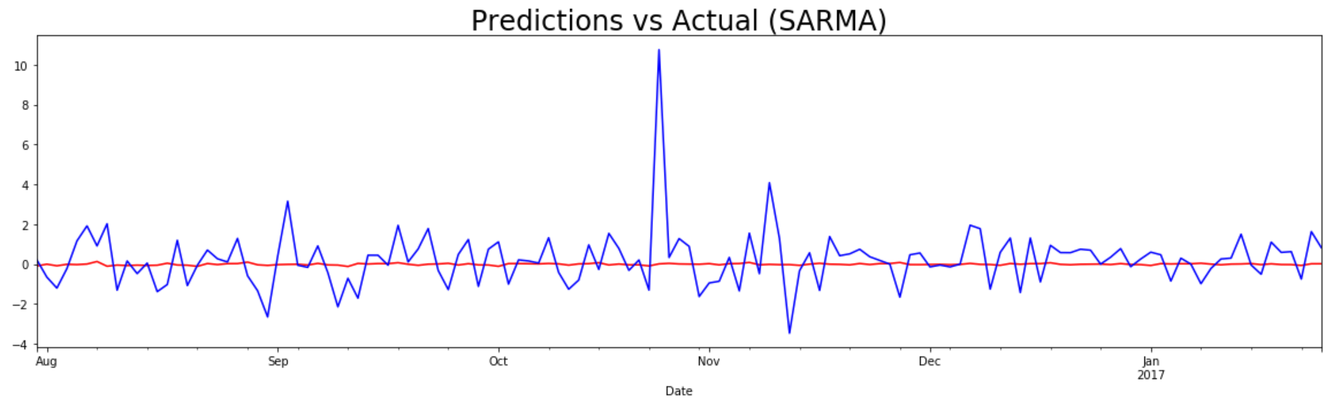
	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0160	0.013	1.197	0.231	-0.010	0.042
ret_FORTUM	0.0610	0.012	5.066	0.000	0.037	0.085
ret_METSO	0.0773	0.009	8.526	0.000	0.060	0.095
ret_STERV	0.6885	0.010	71.605	0.000	0.670	0.707
ar.L1.ret_UPM	0.5779	0.058	9.962	0.000	0.464	0.692
ma.L1.ret_UPM	-0.7119	0.050	-14.220	0.000	-0.810	-0.614

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.7303	+0.0000j	1.7303	0.0000
MA.1	1.4047	+0.0000j	1.4047	0.0000

In [28]:

```
#SARMA
end = "2017-01-25"
model_ret_sarma = SARIMAX(train.ret_UPM[1:], order = (3,0,4), seasonal_order = (3,0,2,5))
results_ret_sarma = model_ret_sarma.fit()

df_pred_sarma = results_ret_sarma.predict(start = start, end = end)

df_pred_sarma[start:end].plot(figsize = (20,5), color = "red")
test.ret_UPM[start:end].plot(color = "blue")
plt.title("Predictions vs Actual (SARMA)", size = 24)
plt.show()
```



In [29]:

```
results_ret_sarma .summary()
```

Out[29]:

SARIMAX Results

Dep. Variable:	ret_UPM	**No. Observations:**	4318
Model:	SARIMAX(3, 0, 4)x(3, 0, [1, 2], 5)	**Log Likelihood**	-9405.689
Date:	Wed, 23 Sep 2020	**AIC**	18837.378
Time:	00:34:02	**BIC**	18920.195
Sample:	01-11-2000	**HQIC**	18866.621

- 07-28-2016

Covariance Type:	opg

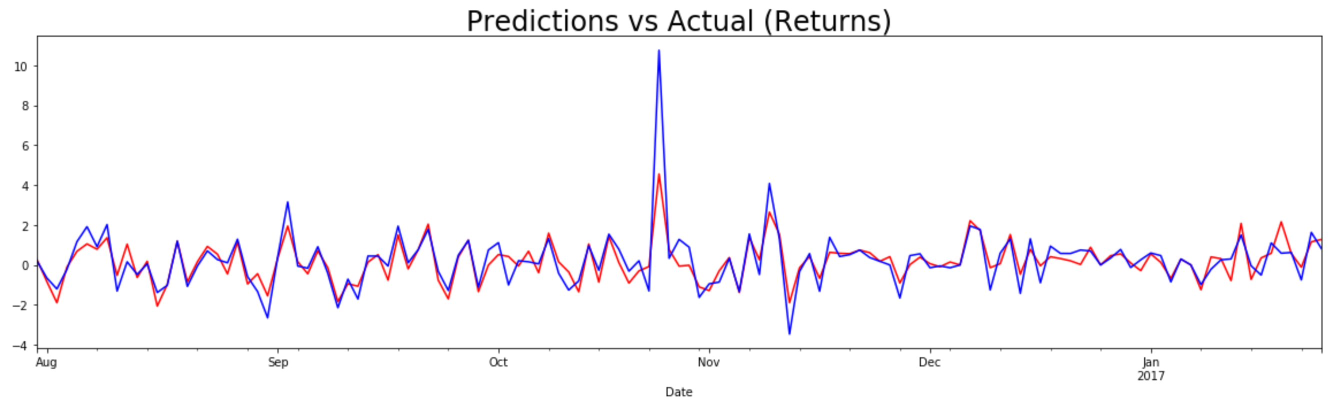
	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.3341	0.290	-1.151	0.250	-0.903	0.235
ar.L2	0.5830	0.133	4.370	0.000	0.322	0.844
ar.L3	0.5775	0.221	2.616	0.009	0.145	1.010
ma.L1	0.3634	0.291	1.250	0.211	-0.206	0.933
ma.L2	-0.5859	0.131	-4.469	0.000	-0.843	-0.329
ma.L3	-0.6194	0.220	-2.815	0.005	-1.051	-0.188
ma.L4	-0.0009	0.021	-0.042	0.967	-0.042	0.041
ar.S.L5	-0.7339	0.060	-12.255	0.000	-0.851	-0.617
ar.S.L10	-0.9355	0.060	-15.522	0.000	-1.054	-0.817
ar.S.L15	0.0052	0.016	0.315	0.753	-0.027	0.037
ma.S.L5	0.7485	0.060	12.534	0.000	0.631	0.865
ma.S.L10	0.9464	0.059	15.916	0.000	0.830	1.063
sigma2	4.5071	0.060	75.681	0.000	4.390	4.624

Ljung-Box (Q):	58.23	**Jarque-Bera (JB):**	2045.95
Prob(Q):	0.03	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.92	**Skew:**	0.11
Prob(H) (two-sided):	0.14	**Kurtosis:**	6.36

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

In [30]:

```
#ARMA
end = "2017-01-25"
model_ret_sarimax = SARIMAX(train.ret_UPM[1:], exog =
train[["ret_FORTUM","ret_METSO","ret_STERV"]][1:], order = (3,0,4),
                          seasonal_order=(3,0,2,5))
results_ret_sarimax= model_ret_sarimax.fit()

df_pred_sarimax = results_ret_sarimax.predict(start = start, end = end,
                                          exog =test[["ret_FORTUM","ret_METSO","ret_STERV"]][start:
nd])

df_pred_sarimax[start:end].plot(figsize = (20,5), color = "red")
test.ret_UPM[start:end].plot(color = "blue")
plt.title("Predictions vs Actual (Returns)", size = 24)
plt.show()
```



In [31]:

```
results_ret_sarimax.summary()
```

Out[31]:

SARIMAX Results

Dep. Variable:	ret_UPM	**No. Observations:**	4318
Model:	SARIMAX(3, 0, 4)x(3, 0, [1, 2], 5)	**Log Likelihood**	-7187.575
Date:	Wed, 23 Sep 2020	**AIC**	14407.150
Time:	00:34:30	**BIC**	14509.079
Sample:	01-11-2000	**HQIC**	14443.141
	- 07-28-2016		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ret_FORTUM	0.0606	0.010	6.230	0.000	0.042	0.080
ret_METSO	0.0774	0.007	11.731	0.000	0.064	0.090
ret_STERV	0.6883	0.007	105.271	0.000	0.675	0.701
ar.L1	-0.3903	0.085	-4.583	0.000	-0.557	-0.223
ar.L2	-0.4145	0.069	-5.989	0.000	-0.550	-0.279
ar.L3	-0.6056	0.062	-9.727	0.000	-0.728	-0.484
ma.L1	0.2661	0.085	3.129	0.002	0.099	0.433
ma.L2	0.2624	0.072	3.620	0.000	0.120	0.405
ma.L3	0.4796	0.067	7.141	0.000	0.348	0.611
ma.L4	-0.1395	0.017	-8.294	0.000	-0.172	-0.107
ar.S.L5	0.1712	47.049	0.004	0.997	-92.042	92.385
ar.S.L10	0.2327	24.997	0.009	0.993	-48.760	49.226
ar.S.L15	0.0114	1.355	0.008	0.993	-2.645	2.668
ma.S.L5	-0.2349	47.049	-0.005	0.996	-92.449	91.979
ma.S.L10	-0.2268	27.989	-0.008	0.994	-55.084	54.631
sigma2	1.6341	0.018	92.488	0.000	1.599	1.669

Ljung-Box (Q):	34.04	**Jarque-Bera (JB):**	7533.54
Prob(Q):	0.73	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.44	**Skew:**	0.25
Prob(H) (two-sided):	0.00	**Kurtosis:**	9.45

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

In [32]:

```
#Auto Arima
model_auto = auto_arima(train.ret_UPM[1:], exogenous = train[['ret_FORTUM', 'ret_METSO', 'ret_STERV
']][1:],
                        m = 5, max_p = 5, max_q = 5, max_P = 5, max_Q = 5)
```

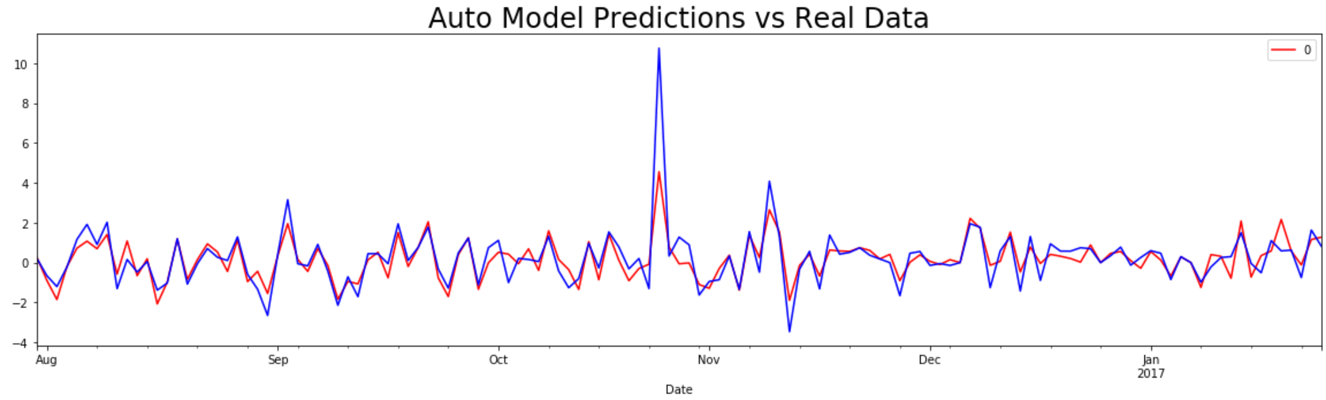
In [33]:

```
df_auto_pred = pd.DataFrame(model_auto.predict(n_periods = len(test[start:end]),
                            exogenous = test[['ret_FORTUM', 'ret_METSO', 'ret_STERV']][start:end]),
                            index = test[start:end].index)
```

In [34]:

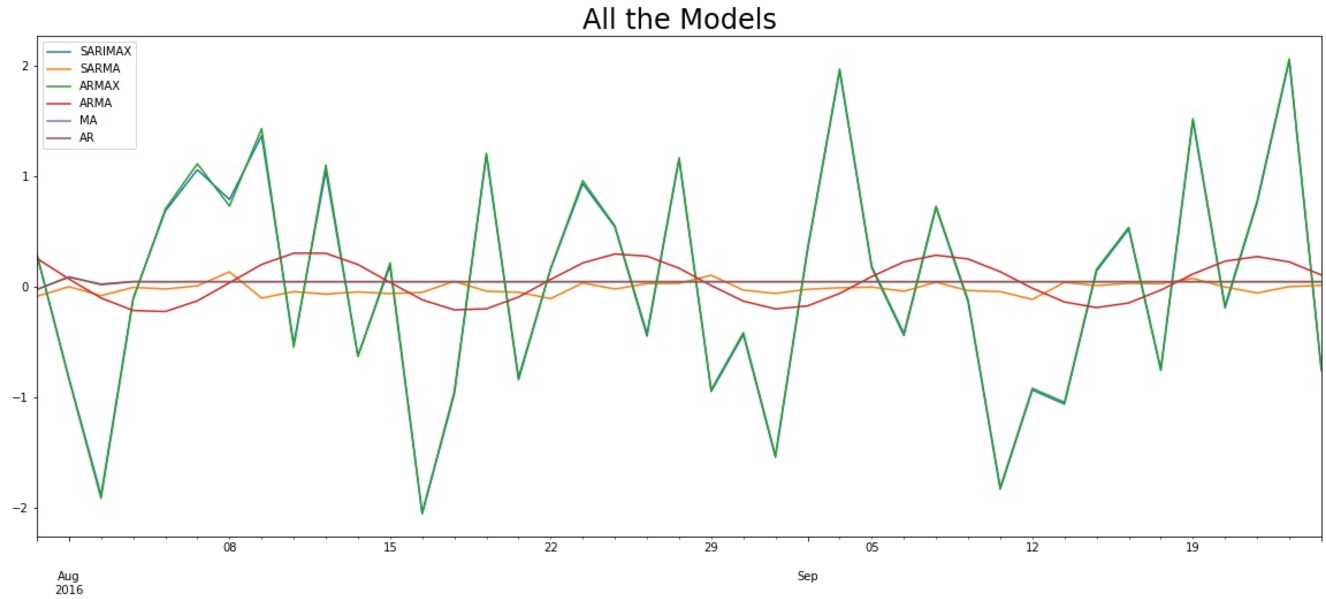
```
df_auto_pred.plot(figsize = (20,5), color = "red")
test.ret_UPM[start:end].plot(color = "blue")
```

```
test.ret_UPM[start:end].plot(color = "blue")
plt.title("Auto Model Predictions vs Real Data", size = 24)
plt.show()
```



In [42]:

```
#Comparing all the models
end="2016-09-25"

df_pred_sarimax[start:end].plot(figsize=(20,8) ,legend= True)
df_pred_sarma[start:end].plot(legend =True)
df_pred_armax[start:end].plot(legend=True)
df_pred_arma[start:end].plot(legend=True)
df_pred_ma[start:end].plot(legend=True )
df_pred_ar[start:end].plot(legend=True)
plt.legend(['SARIMAX','SARMA','ARMAX','ARMA','MA','AR'])
plt.title("All the Models", size = 24)
plt.show()
```



Forecasting Volatility(GARCH)

In [43]:

```
mod_garch = arch_model(df.ret_UPM[1:], vol = "GARCH", p = 1, q = 1, mean = "constant", dist =
"Normal")
res_garch = mod_garch.fit(last_obs = start, update_freq = 10)
```

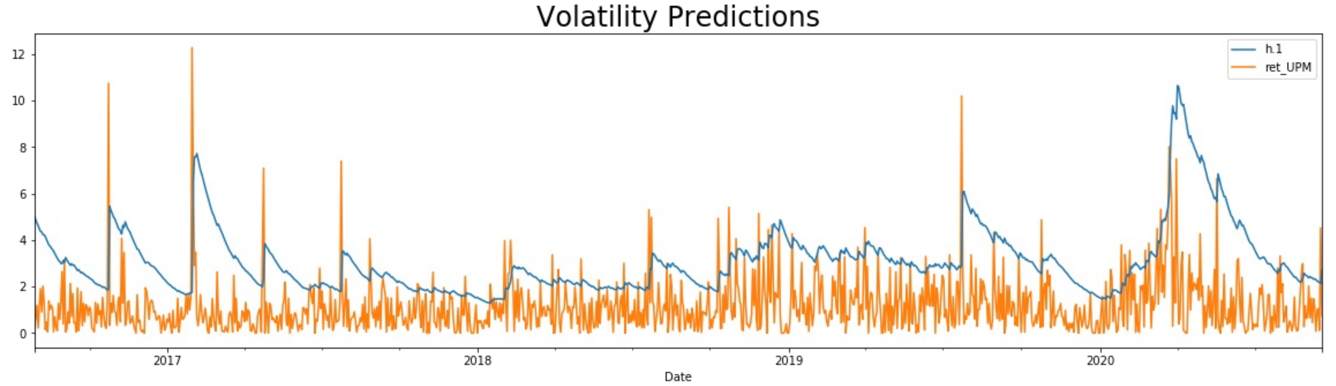
```
Iteration:      10,   Func. Count:      74,   Neg. LLF: 9053.804730489617
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 9053.342561391617
            Iterations: 16
            Function evaluations: 113
            Gradient evaluations: 16
```

In [44]:

```
pred_garch = res_garch.forecast(horizon = 1, align = 'target')
```

In [45]:

```
pred_garch.residual_variance[start:].plot(figsize = (20,5),legend=True, zorder = 2)
test.ret_UPM.abs().plot(legend=True, zorder = 1)
plt.title("Volatility Predictions", size = 24)
plt.show()
```



In [46]:

```
pred_garch = res_garch.forecast(horizon = 100, align = 'target')
pred_garch.residual_variance[-1:]
```

Out[46]:

	h.001	h.002	h.003	h.004	h.005	h.006	h.007	h.008	h.009	h.010	...	h.091	h.092	h.09:
Date														
2020-09-17	2.710105	2.144142	2.212408	2.243078	2.286316	2.358164	2.396006	2.394965	2.463345	2.472177	...	5.39751	5.483715	5.60:

1 rows × 100 columns

Multivariate Regression Model (VAR)

In [48]:

```
from statsmodels.tsa.api import VAR
```

In [49]:

```
df_returns = df[['ret_UPM', 'ret_FORTUM', 'ret_METSO', 'ret_STERV']][1:]
```

In [51]:

```
model_var_ret = VAR(df_returns)
model_var_ret.select_order(20)
results_var_ret = model_var_ret.fit(ic = 'aic')
```

In [52]:

```
results_var_ret.summary()
```

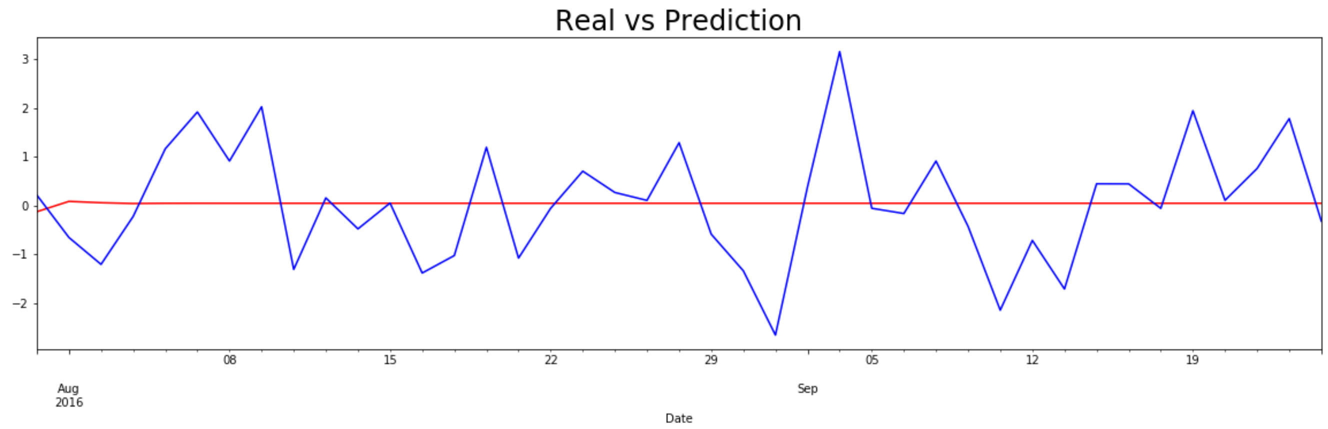
Out[52]:

```
  Summary of Regression Results
```

```
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 23, Sep, 2020
Time:                     01:21:46
--------------------------------------------------------------------
No. of Equations:         4.00000    BIC:                    4.67010
Nobs:                     5396.00    HQIC:                   4.64147
Log likelihood:          -43071.6    FPE:                    102.116
AIC:                      4.62611    Det(Omega_mle):         101.438
--------------------------------------------------------------------
Results for equation ret_UPM
================================================================================
                   coefficient       std. error           t-stat            prob
--------------------------------------------------------------------------------
const                 0.048587         0.027985            1.736           0.083
L1.ret_UPM           -0.036065         0.022767           -1.584           0.113
L1.ret_FORTUM         0.004324         0.017895            0.242           0.809
L1.ret_METSO          0.001358         0.012235            0.111           0.912
L1.ret_STERV          0.073638         0.021241            3.467           0.001
L2.ret_UPM           -0.071964         0.022788           -3.158           0.002
L2.ret_FORTUM        -0.002111         0.017890           -0.118           0.906
L2.ret_METSO          0.022510         0.012231            1.840           0.066
L2.ret_STERV          0.038362         0.021215            1.808           0.071
================================================================================

Results for equation ret_FORTUM
================================================================================
                   coefficient       std. error           t-stat            prob
--------------------------------------------------------------------------------
const                 0.070041         0.023388            2.995           0.003
L1.ret_UPM           -0.005919         0.019027           -0.311           0.756
L1.ret_FORTUM        -0.025732         0.014955           -1.721           0.085
L1.ret_METSO          0.025574         0.010225            2.501           0.012
L1.ret_STERV          0.013846         0.017751            0.780           0.435
L2.ret_UPM            0.029485         0.019044            1.548           0.122
L2.ret_FORTUM        -0.024661         0.014951           -1.649           0.099
L2.ret_METSO          0.015759         0.010222            1.542           0.123
L2.ret_STERV         -0.034866         0.017730           -1.966           0.049
================================================================================

Results for equation ret_METSO
================================================================================
                   coefficient       std. error           t-stat            prob
--------------------------------------------------------------------------------
const                 0.057809         0.035584            1.625           0.104
L1.ret_UPM            0.068788         0.028948            2.376           0.017
L1.ret_FORTUM         0.047895         0.022753            2.105           0.035
L1.ret_METSO         -0.036285         0.015557           -2.332           0.020
L1.ret_STERV         -0.001674         0.027008           -0.062           0.951
L2.ret_UPM            0.052415         0.028975            1.809           0.070
L2.ret_FORTUM        -0.030833         0.022748           -1.355           0.175
L2.ret_METSO          0.008428         0.015553            0.542           0.588
L2.ret_STERV         -0.026757         0.026976           -0.992           0.321
================================================================================

Results for equation ret_STERV
================================================================================
                   coefficient       std. error           t-stat            prob
--------------------------------------------------------------------------------
const                 0.033060         0.030463            1.085           0.278
L1.ret_UPM            0.113364         0.024782            4.574           0.000
L1.ret_FORTUM        -0.007782         0.019479           -0.400           0.690
L1.ret_METSO          0.007202         0.013318            0.541           0.589
L1.ret_STERV         -0.058029         0.023121           -2.510           0.012
L2.ret_UPM            0.018427         0.024805            0.743           0.458
L2.ret_FORTUM        -0.002466         0.019474           -0.127           0.899
L2.ret_METSO          0.009062         0.013314            0.681           0.496
L2.ret_STERV         -0.006503         0.023093           -0.282           0.778
================================================================================

Correlation matrix of residuals
               ret_UPM  ret_FORTUM  ret_METSO  ret_STERV
ret_UPM       1.000000    0.332743   0.407899   0.798908
ret_FORTUM    0.332743    1.000000   0.342841   0.347109
ret_METSO     0.407899    0.342841   1.000000   0.428037
ret_STERV     0.798908    0.347109   0.428037   1.000000
```

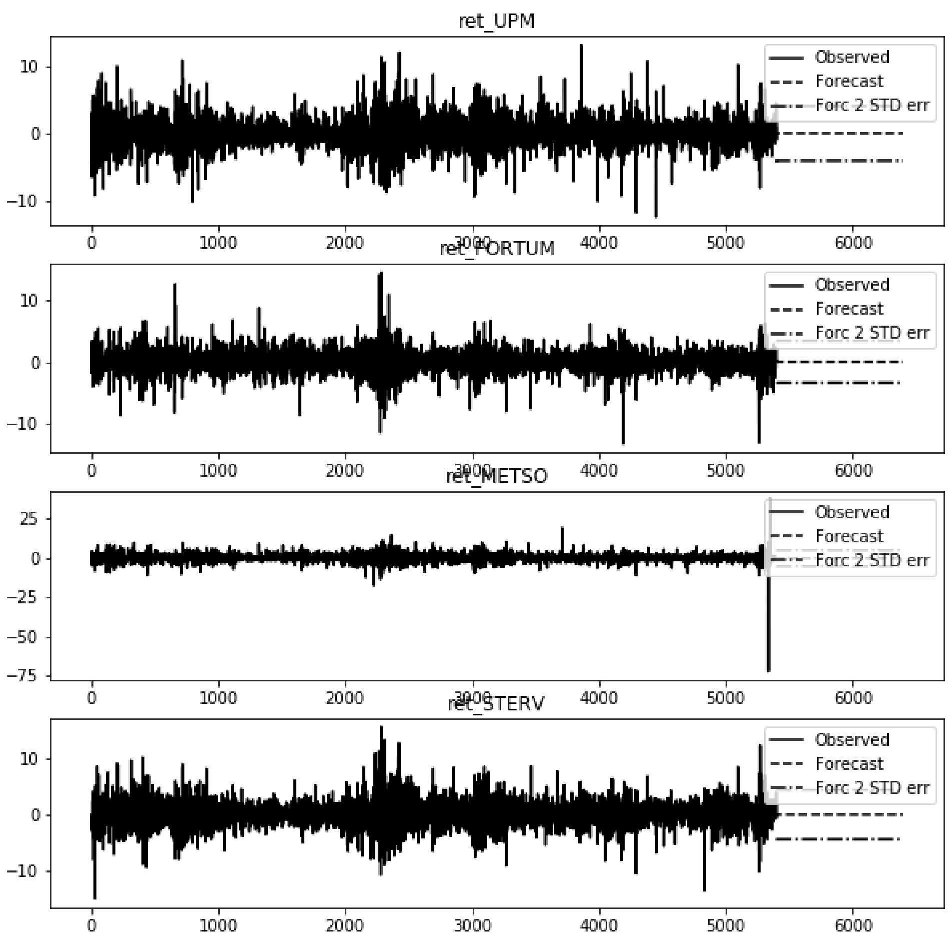
In [54]:

```
lag_order_ret = results_var_ret.k_ar
var_pred_ret = results_var_ret.forecast(df_returns.values[-lag_order_ret:], len(test[start:end]))

df_ret_pred = pd.DataFrame(data = var_pred_ret, index =test[start:end].index,
                                 columns = test[start:end].columns[4:8])

df_ret_pred.ret_UPM[start:end].plot(figsize = (20,5), color = "red")

test.ret_UPM[start:end].plot(color = "blue")
plt.title("Real vs Prediction", size = 24)
plt.show()
```



In [59]:

```
results_var_ret.plot_forecast(1000)
plt.show()
```



In [ ]:

```

```