```
# The effect of first Official Announcement from Volkswagen to own 49.9%(2009-12-09)
Porsche,Second Official Announcement to own 51.1%(2012-07-05) of Porsche and finally the
Volksagen Diesel Scandal(2015-09-20).
```

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import yfinance as yf
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
#Importing Data from Yahoo Finance
df=yf.download(tickers="VOW3.DE, PAH3.DE, BMW.DE", interval = "1d", group_by = 'ticker',auto_adjust
= True, treads = True)
```

```
[*********************100%***********************]  3 of 3 completed
```

In [3]:

```
#some important Dates and Announcements from Volkswagen(VW)

#Start Date
start='2009-01-01'

# first official Announcement (VW announced the own 49.9% of Porsche)
f_ann='2009-12-09'

#Second Announcement(VW announced the purchase of the remaining 51.1% of Porsche)
s_ann='2012-07-05'

# End Date
end='2014-01-01'

#Diesel Scandal emission
d_sca='2015-09-20'
```

Pre-processing Data,Closing prices,Returns,Squared Returns and Volume.

In [4]:

```
#Exacting Closing Prices
df['vog'] = df['VOW3.DE'].Close
df['por'] = df['PAH3.DE'].Close
df['bmw'] = df['BMW.DE'].Close
```

In [5]:

```
# Creating Returns
df['ret_vog'] = df['vog'].pct_change(1).mul(100)
df['ret_por'] = df['por'].pct_change(1).mul(100)
df['ret_bmw'] = df['bmw'].pct_change(1).mul(100)
```

In [6]:

```
# Creating Squared Returns(Volatility)
df['sq_vog'] = df.ret_vog.mul(df.ret_vog)
df['sq_por'] = df.ret_por.mul(df.ret_por)
df['sq_bmw'] = df.ret_bmw.mul(df.ret_bmw)
```

In [10]:

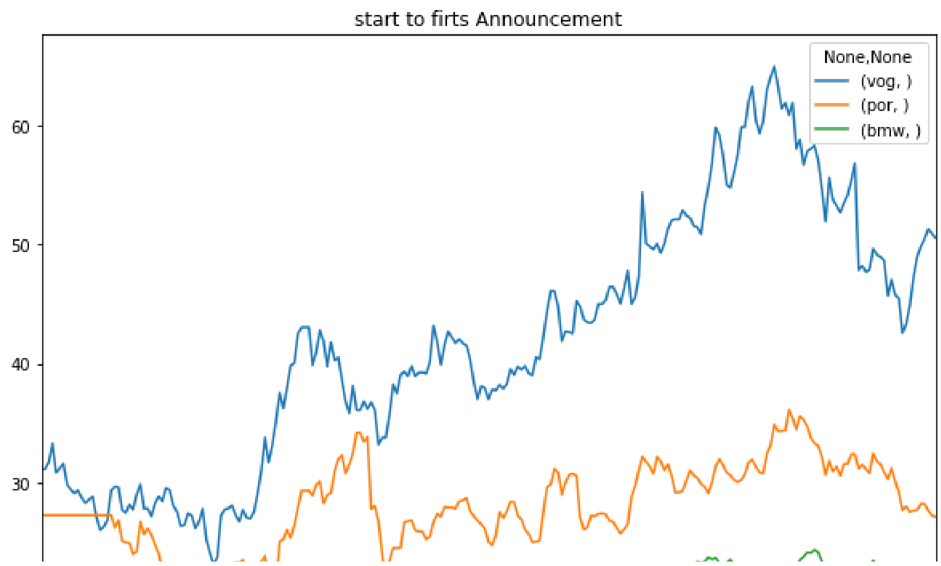
```
#making our frequency Daily for Business working days and filling NA value with bfill or ffill met
hod
df=df.asfreq(freq='b')
df=df.fillna(method='bfill')
```

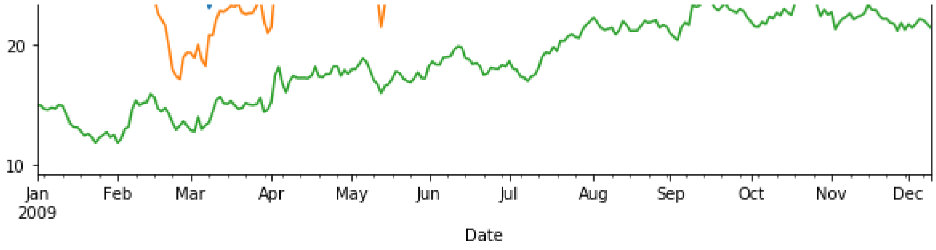
In [12]:

```
# Deleting some surplus data
del df['VOW3.DE'],df['PAH3.DE'],df['BMW.DE']
```

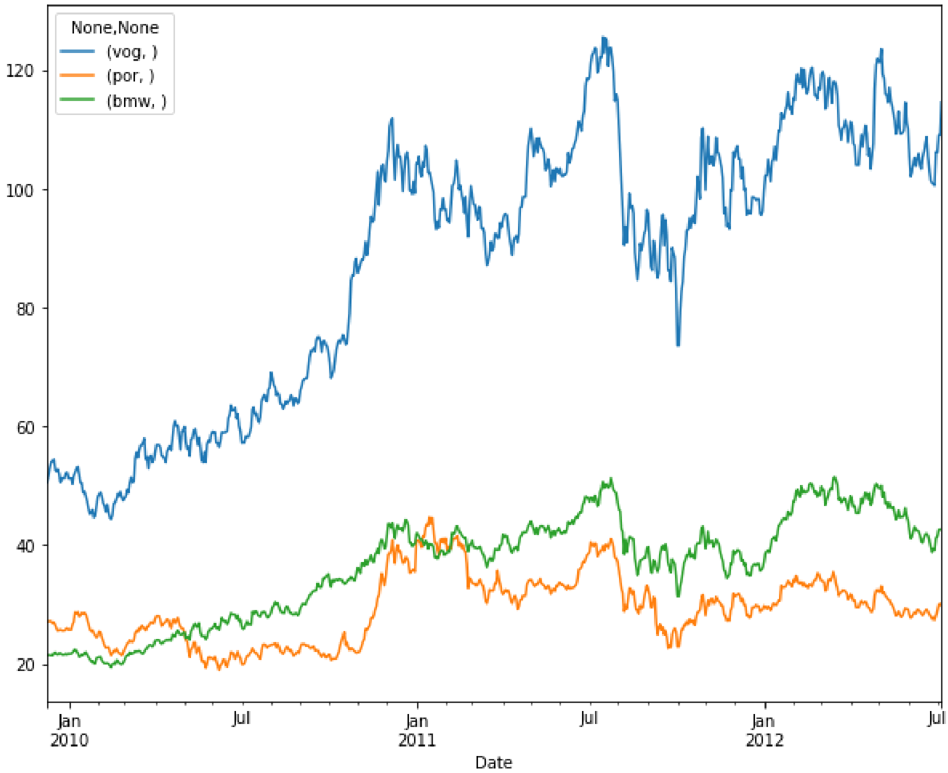
In [39]:

```
#ploting our graphs base on the important annoucements and disiel scandal
df[['vog','por','bmw']][start:end].plot(figsize= (10,8),title='start to end')
df[['vog','por','bmw']][start:f_ann].plot(figsize= (10,8),title='start to firts Announcement')
df[['vog','por','bmw']][f_ann:s_ann].plot(figsize= (10,8),title='first announcement to second
Announcement')
df[['vog','por','bmw']][s_ann:end].plot(figsize= (10,8),title='second announcement to end')
df[['vog','por','bmw']][end:d_sca].plot(figsize= (10,8),title='end to disiel scandal')
df[['vog','por','bmw']][d_sca:].plot(figsize= (10,8),title='start to firts Announcement')
```



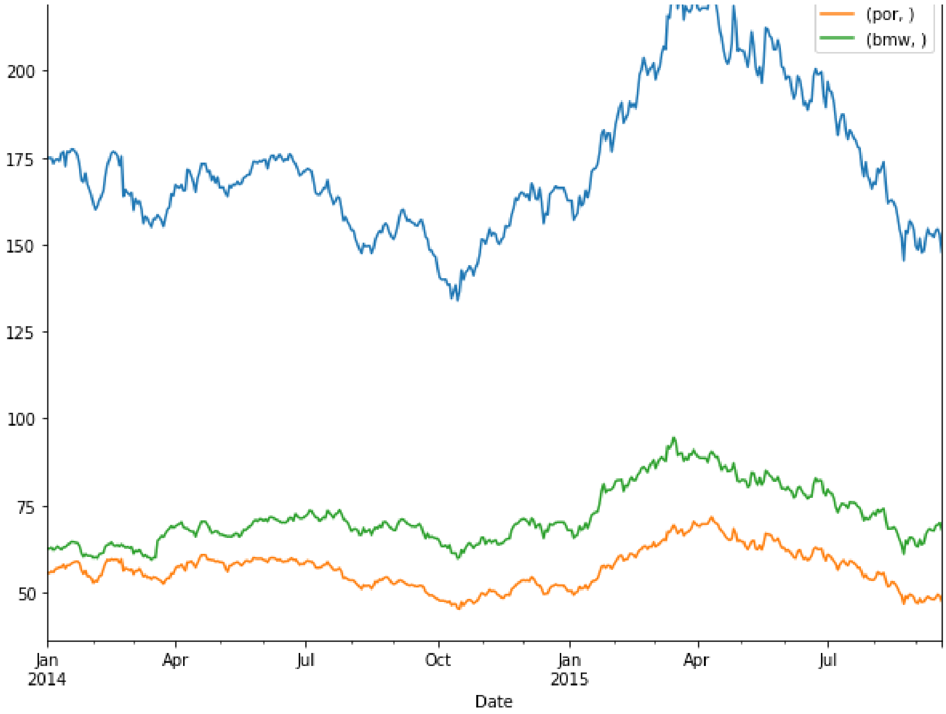


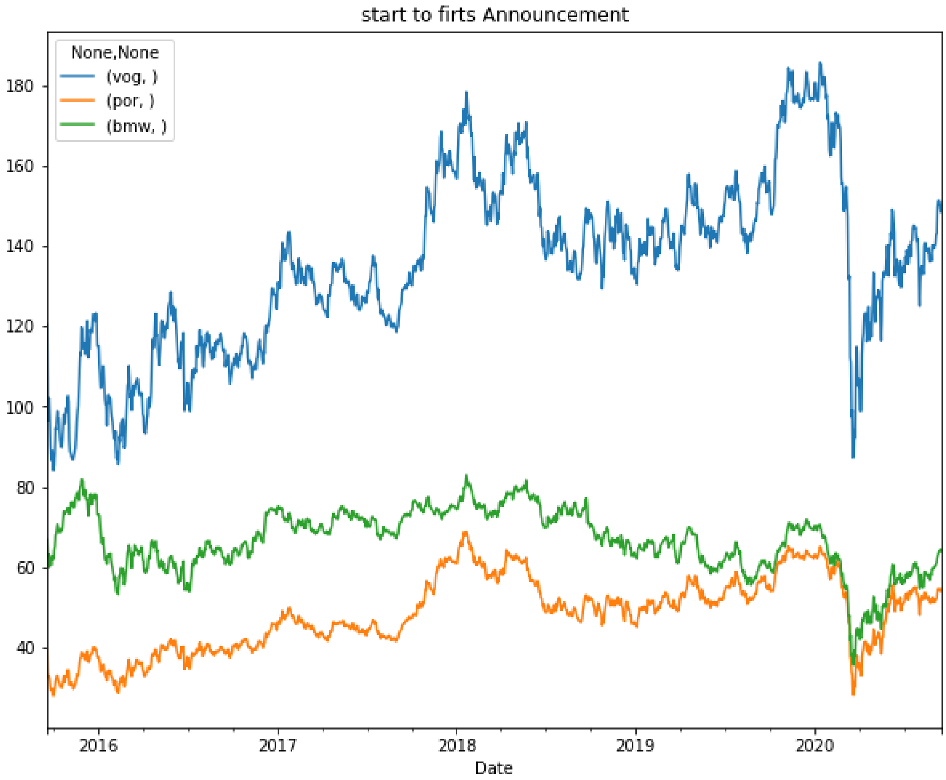
first announcement to second Announcement



second announcement to end

end to disiel scandal





In [42]:

```
#correlations for closing prices
import seaborn as sns
df[['vog','por','bmw']].corr()
```

Out[42]:

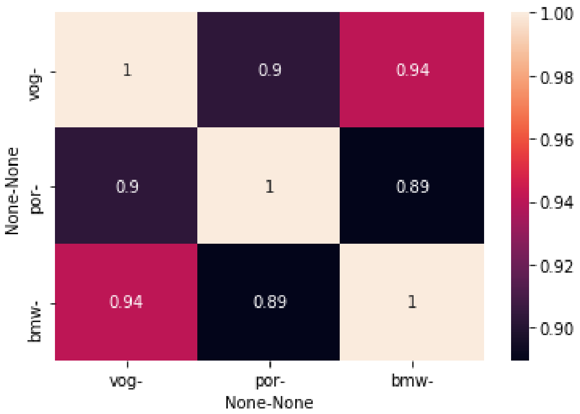
	vog	por	bmw
vog	1.000000	0.902956	0.940954
por	0.902956	1.000000	0.889573
bmw	0.940954	0.889573	1.000000

In [43]:

```
sns.heatmap(df[['vog','por','bmw']].corr(),annot=True)
```

Out[43]:

<matplotlib.axes._subplots.AxesSubplot at 0x16168335d08>



In [32]:

```
print('Correlation among manufacturers from ' + str(start_date) + ' to ' + str(end_date)+ '\n')
print('Volkswagen and Porsche correlation: \t'+ str(df['vol'][start_date:end_date].corr(df['por'][s
tart_date:end_date])))
print('Volkswagen and BMW correlation: \t'+ str(df['vol'][start_date:end_date].corr(df['bmw'][start
_date:end_date])))
print('Porsche and BMW correlation: \t\t'+ str(df['por'][start_date:end_date].corr(df['bmw'][start_
date:end_date])))
```

Out[32]:

<matplotlib.axes._subplots.AxesSubplot at 0x1616be7cc48>

In [55]:

```
#Anaylsing correlation from start to end,important announcement dates and diesel scandal
annoucement
print('Correlation among manufacturers from :\t' + str(start) + ' to ' + str(end)+ '\n')
print('Volkswagen and Porsche correlation:\t'+ str(df['vog'][start:end].corr(df['por'][start:end]))
)
print('Volkswagen and BMW correlation: \t'+ str(df['vog'][start:end].corr(df['bmw'][start:end])))
print('Porsche and BMW correlation: \t\t'+ str(df['por'][start:end].corr(df['bmw'][start:end])))
```

Correlation among manufacturers from : 2009-01-01 to 2014-01-01

Volkswagen and Porsche correlation: 0.8386338628707073

```
Volkswagen and BMW correlation:  0.9827579671118887
Porsche and BMW correlation:   0.8089641486724635
```

In [57]:

```python
print('Correlation among manufacturers from :\t' + str(start) + ' to ' + str(f_ann)+ '\n')
print('Volkswagen and Porsche correlation:\t'+ str(df['vog'][start:f_ann].corr(df['por'][start:f_ann])))
print('Volkswagen and BMW correlation: \t'+ str(df['vog'][start:f_ann].corr(df['bmw'][start:f_ann])))
print('Porsche and BMW correlation: \t\t'+ str(df['por'][start:f_ann].corr(df['bmw'][start:f_ann])))
```

```
Correlation among manufacturers from : 2009-01-01 to 2009-12-09

Volkswagen and Porsche correlation: 0.7785123807487967
Volkswagen and BMW correlation:  0.9263430965312215
Porsche and BMW correlation:   0.7345779243010784
```

In [58]:

```python
print('Correlation among manufacturers from :\t' + str(f_ann) + ' to ' + str(s_ann)+ '\n')
print('Volkswagen and Porsche correlation:\t'+ str(df['vog'][f_ann:s_ann].corr(df['por'][f_ann:s_ann])))
print('Volkswagen and BMW correlation: \t'+ str(df['vog'][f_ann:s_ann].corr(df['bmw'][f_ann:s_ann])))
print('Porsche and BMW correlation: \t\t'+ str(df['por'][f_ann:s_ann].corr(df['bmw'][f_ann:s_ann])))
```

```
Correlation among manufacturers from : 2009-12-09 to 2012-07-05

Volkswagen and Porsche correlation: 0.7422114347356783
Volkswagen and BMW correlation:  0.9795942993967812
Porsche and BMW correlation:   0.7035985449323026
```

In [59]:

```python
print('Correlation among manufacturers from :\t' + str(s_ann) + ' to ' + str(end)+ '\n')
print('Volkswagen and Porsche correlation:\t'+ str(df['vog'][s_ann:end].corr(df['por'][s_ann:end])))
print('Volkswagen and BMW correlation: \t'+ str(df['vog'][s_ann:end].corr(df['bmw'][s_ann:end])))
print('Porsche and BMW correlation: \t\t'+ str(df['por'][s_ann:end].corr(df['bmw'][s_ann:end])))
```

```
Correlation among manufacturers from : 2012-07-05 to 2014-01-01

Volkswagen and Porsche correlation: 0.9405236894284832
Volkswagen and BMW correlation:  0.9284447118744797
Porsche and BMW correlation:   0.9494111752233421
```

In [60]:

```python
print('Correlation among manufacturers from :\t' + str(end) + ' to ' + str(d_sca)+ '\n')
print('Volkswagen and Porsche correlation:\t'+ str(df['vog'][end:d_sca].corr(df['por'][end:d_sca])))
print('Volkswagen and BMW correlation: \t'+ str(df['vog'][end:d_sca].corr(df['bmw'][end:d_sca])))
print('Porsche and BMW correlation: \t\t'+ str(df['por'][end:d_sca].corr(df['bmw'][end:d_sca])))
```

```
Correlation among manufacturers from : 2014-01-01 to 2015-09-20

Volkswagen and Porsche correlation: 0.9421376075139968
Volkswagen and BMW correlation:  0.8912208007790657
Porsche and BMW correlation:   0.8045871574266078
```

In [61]:

```python
print('Correlation among manufacturers from :\t' + str(d_sca) + ' to ' + str(df.index[-1])+ '\n')
print('Volkswagen and Porsche correlation:\t'+ str(df['vog'][d_sca:].corr(df['por'][d_sca:])))
print('Volkswagen and BMW correlation: \t'+ str(df['vog'][d_sca:].corr(df['bmw'][d_sca:])))
print('Porsche and BMW correlation: \t\t'+ str(df['por'][d_sca:].corr(df['bmw'][d_sca:])))
```

```
Correlation among manufacturers from : 2015-09-20 to 2020-09-15 00:00:00

Volkswagen and Porsche correlation: 0.9790570450497561
Volkswagen and BMW correlation:  0.32173659309314
Porsche and BMW correlation:   0.3339095623502678
```

Time to fit in the best models for our analysis of the different prices,with Vog and Por and BMW exogenous variables respectively.Our annoucement dates will also be included to draw an explicit conclusion.

In [62]:

```python
from pmdarima.arima import auto_arima,OCSBTest
```

In [65]:

```python
# bestfit for Vog
mod_start_first_annVog = auto_arima(df.vog[start:f_ann], exogenous = df[['por','bmw']][start:f_ann
],
                            m = 5, max_p = 5, max_q = 5)
mod_first_second_annVog = auto_arima(df.vog[f_ann:s_ann], exogenous = df[['por','bmw']]
[f_ann:s_ann],
                            m = 5, max_p = 5, max_q = 5)
mod_second_end_annVog = auto_arima(df.vog[s_ann:end], exogenous = df[['por','bmw']][s_ann:end],
                            m = 5, max_p = 5, max_q = 5)
mod_end_scandal_annVog= auto_arima(df.vog[end:d_sca], exogenous = df[['por','bmw']][end:d_sca],
                            m = 5, max_p = 5, max_q = 5)
mod_scandal_today_annVog =  auto_arima(df.vog[d_sca:], exogenous = df[['por','bmw']][d_sca:],
                            m = 5, max_p = 5, max_q = 5)
```

In [66]:

```python
# Analysing summary statistics for Vog
mod_start_first_annVog.summary()
```

Out[66]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	245
Model:	SARIMAX(1, 0, 0)	**Log Likelihood**	-435.343
Date:	Wed, 16 Sep 2020	**AIC**	880.686
Time:	14:56:21	**BIC**	898.192
Sample:	01-01-2009	**HQIC**	887.736
	- 12-09-2009		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.3400	0.262	1.296	0.195	-0.174	0.854
por	0.4034	0.077	5.229	0.000	0.252	0.555
bmw	0.6617	0.229	2.886	0.004	0.212	1.111
ar.L1	0.9812	0.013	77.848	0.000	0.956	1.006
sigma2	2.0188	0.112	18.074	0.000	1.800	2.238

Ljung-Box (Q):	47.55	**Jarque-Bera (JB):**	263.95
Prob(Q):	0.19	**Prob(JB):**	0.00
Heteroskedasticity (H):	2.44	**Skew:**	-0.48
Prob(H) (two-sided):	0.00	**Kurtosis:**	7.99

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

In [67]:

```
mod_first_second_annVog.summary()
```

Out[67]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	672
Model:	SARIMAX(0, 1, 0)	**Log Likelihood**	-1105.059
Date:	Wed, 16 Sep 2020	**AIC**	2216.118
Time:	14:57:01	**BIC**	2229.644
Sample:	12-09-2009	**HQIC**	2221.357
	- 07-05-2012		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
por	0.8203	0.064	12.833	0.000	0.695	0.946
bmw	1.5349	0.072	21.267	0.000	1.393	1.676
sigma2	1.5776	0.064	24.528	0.000	1.452	1.704

Ljung-Box (Q):	33.73	**Jarque-Bera (JB):**	112.32
Prob(Q):	0.75	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.65	**Skew:**	0.29
Prob(H) (two-sided):	0.00	**Kurtosis:**	4.92

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

In [68]:

```
mod_second_end_annVog .summary()
```

Out[68]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	390
Model:	SARIMAX(0, 1, 0)	**Log Likelihood**	-622.244
Date:	Wed, 16 Sep 2020	**AIC**	1250.488
Time:	14:58:00	**BIC**	1262.378
Sample:	07-05-2012	**HQIC**	1255.202
	- 01-01-2014		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
por	1.7837	0.059	30.369	0.000	1.669	1.899
bmw	0.8498	0.091	9.381	0.000	0.672	1.027
sigma2	1.4352	0.070	20.461	0.000	1.298	1.573

Ljung-Box (Q):	28.08	**Jarque-Bera (JB):**	103.80

Prob(Q):	0.92	Prob(JB):	0.00
Heteroskedasticity (H):	0.78	Skew:	-0.39
Prob(H) (two-sided):	0.15	Kurtosis:	5.41

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

In [69]:

```
mod_end_scandal_annVog.summary()
```

Out[69]:

SARIMAX Results

Dep. Variable:	y	No. Observations:	448
Model:	SARIMAX(0, 1, 0)x(1, 0, 0, 5)	Log Likelihood	-686.483
Date:	Wed, 16 Sep 2020	AIC	1380.966
Time:	14:58:34	BIC	1397.376
Sample:	01-01-2014	HQIC	1387.436
	- 09-18-2015		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
por	2.1800	0.069	31.708	0.000	2.045	2.315
bmw	0.6491	0.063	10.349	0.000	0.526	0.772
ar.S.L5	-0.0994	0.046	-2.150	0.032	-0.190	-0.009
sigma2	1.2630	0.068	18.609	0.000	1.130	1.396

Ljung-Box (Q):	35.77	Jarque-Bera (JB):	31.21
Prob(Q):	0.66	Prob(JB):	0.00
Heteroskedasticity (H):	1.30	Skew:	-0.20
Prob(H) (two-sided):	0.11	Kurtosis:	4.23

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

In [70]:

```
mod_scandal_today_annVog.summary()
```

Out[70]:

SARIMAX Results

Dep. Variable:	y	No. Observations:	1302
Model:	SARIMAX(0, 1, 3)x(1, 0, [1], 5)	Log Likelihood	-1935.900
Date:	Wed, 16 Sep 2020	AIC	3887.799
Time:	14:59:30	BIC	3929.167
Sample:	09-21-2015	HQIC	3903.320
	- 09-15-2020		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
por	2.3551	0.028	84.519	0.000	2.300	2.410

bmw	0.4171	0.033	12.678	0.000	0.353	0.482
ma.L1	-0.0610	0.024	-2.570	0.010	-0.108	-0.014
ma.L2	-0.0637	0.023	-2.766	0.006	-0.109	-0.019
ma.L3	-0.0494	0.024	-2.047	0.041	-0.097	-0.002
ar.S.L5	-0.7765	0.163	-4.767	0.000	-1.096	-0.457
ma.S.L5	0.7337	0.172	4.255	0.000	0.396	1.072
sigma2	1.1481	0.029	38.923	0.000	1.090	1.206

Ljung-Box (Q):	25.55	**Jarque-Bera (JB):**	604.54
Prob(Q):	0.96	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.11	**Skew:**	-0.42
Prob(H) (two-sided):	0.27	**Kurtosis:**	6.23

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

In [72]:

```
# bestfit for por,with vog and bmw exogenous variables
mod_start_first_annPor = auto_arima(df.por[start:f_ann], exogenous = df[['vog','bmw']][start:f_ann
],
                            m = 5, max_p = 5, max_q = 5)
mod_first_second_annPor = auto_arima(df.por[f_ann:s_ann], exogenous = df[['vog','bmw']]
[f_ann:s_ann],
                            m = 5, max_p = 5, max_q = 5)
mod_second_end_annPor = auto_arima(df.por[s_ann:end], exogenous = df[['vog','bmw']][s_ann:end],
                            m = 5, max_p = 5, max_q = 5)
mod_end_scandal_annPor= auto_arima(df.por[end:d_sca], exogenous = df[['vog','bmw']][end:d_sca],
                            m = 5, max_p = 5, max_q = 5)
mod_scandal_today_annPor =  auto_arima(df.por[d_sca:], exogenous = df[['vog','bmw']][d_sca:],
                            m = 5, max_p = 5, max_q = 5)
```

In [73]:

```
mod_start_first_annPor.summary()
```

Out[73]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	245
Model:	SARIMAX(2, 0, 1)	**Log Likelihood**	-303.384
Date:	Wed, 16 Sep 2020	**AIC**	620.767
Time:	15:30:31	**BIC**	645.276
Sample:	01-01-2009	**HQIC**	630.637
	- 12-09-2009		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.2280	0.154	1.480	0.139	-0.074	0.530
vog	0.1234	0.044	2.798	0.005	0.037	0.210
bmw	0.7624	0.105	7.244	0.000	0.556	0.969
ar.L1	1.6732	0.211	7.930	0.000	1.260	2.087
ar.L2	-0.7001	0.198	-3.533	0.000	-1.088	-0.312
ma.L1	-0.6167	0.242	-2.552	0.011	-1.090	-0.143
sigma2	0.6900	0.043	16.182	0.000	0.606	0.774

Ljung-Box (Q):	34.57	**Jarque-Bera (JB):**	660.78

Prob(Q):	0.71	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.55	**Skew:**	-0.73
Prob(H) (two-sided):	0.01	**Kurtosis:**	10.91

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

In [74]:

```
mod_first_second_annPor.summary()
```

Out[74]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	672
Model:	SARIMAX(1, 1, 1)	**Log Likelihood**	-581.121
Date:	Wed, 16 Sep 2020	**AIC**	1172.242
Time:	15:31:06	**BIC**	1194.786
Sample:	12-09-2009	**HQIC**	1180.973
	- 07-05-2012		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
vog	0.1719	0.015	11.127	0.000	0.142	0.202
bmw	0.3170	0.044	7.206	0.000	0.231	0.403
ar.L1	0.6598	0.229	2.882	0.004	0.211	1.109
ma.L1	-0.5909	0.245	-2.412	0.016	-1.071	-0.111
sigma2	0.3309	0.010	32.414	0.000	0.311	0.351

Ljung-Box (Q):	33.39	**Jarque-Bera (JB):**	1076.70
Prob(Q):	0.76	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.39	**Skew:**	0.05
Prob(H) (two-sided):	0.01	**Kurtosis:**	9.20

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

In [75]:

```
mod_second_end_annPor.summary()
```

Out[75]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	390
Model:	SARIMAX(0, 1, 0)x(0, 0, [1], 5)	**Log Likelihood**	-210.347
Date:	Wed, 16 Sep 2020	**AIC**	428.694
Time:	15:31:55	**BIC**	444.548
Sample:	07-05-2012	**HQIC**	434.979
	- 01-01-2014		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]

vog	0.2138	0.013	16.876	0.000	0.189	0.239
bmw	0.2392	0.040	6.036	0.000	0.162	0.317
ma.S.L5	-0.1249	0.057	-2.210	0.027	-0.236	-0.014
sigma2	0.1726	0.006	27.851	0.000	0.160	0.185

Ljung-Box (Q):	34.49	**Jarque-Bera (JB):**	613.59
Prob(Q):	0.72	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.57	**Skew:**	0.88
Prob(H) (two-sided):	0.00	**Kurtosis:**	8.90

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

In [76]:

```
mod_end_scandal_annPor.summary()
```

Out[76]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	448
Model:	SARIMAX(0, 1, 0)	**Log Likelihood**	-197.215
Date:	Wed, 16 Sep 2020	**AIC**	400.430
Time:	15:32:38	**BIC**	412.738
Sample:	01-01-2014	**HQIC**	405.283
	- 09-18-2015		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
vog	0.2442	0.008	29.654	0.000	0.228	0.260
bmw	0.1077	0.018	6.117	0.000	0.073	0.142
sigma2	0.1415	0.008	18.276	0.000	0.126	0.157

Ljung-Box (Q):	33.23	**Jarque-Bera (JB):**	23.75
Prob(Q):	0.77	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.84	**Skew:**	0.17
Prob(H) (two-sided):	0.28	**Kurtosis:**	4.07

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

In [78]:

```
mod_scandal_today_annPor.summary()
```

Out[78]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	1302
Model:	SARIMAX(1, 1, 1)x(1, 0, 1, 5)	**Log Likelihood**	-539.606
Date:	Wed, 16 Sep 2020	**AIC**	1093.212
Time:	15:33:28	**BIC**	1129.408
Sample:	09-21-2015	**HQIC**	1106.792

	- 09-15-2020
Covariance Type:	opg

	coef	std err	z	P>|z|	[0.025	0.975]
vog	0.2750	0.004	62.685	0.000	0.266	0.284
bmw	0.1354	0.013	10.582	0.000	0.110	0.160
ar.L1	0.5257	0.236	2.228	0.026	0.063	0.988
ma.L1	-0.5865	0.220	-2.663	0.008	-1.018	-0.155
ar.S.L5	-0.9331	0.050	-18.522	0.000	-1.032	-0.834
ma.S.L5	0.8999	0.060	14.908	0.000	0.782	1.018
sigma2	0.1341	0.003	45.268	0.000	0.128	0.140

Ljung-Box (Q):	47.45	**Jarque-Bera (JB):**	1718.92
Prob(Q):	0.19	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.85	**Skew:**	0.72
Prob(H) (two-sided):	0.00	**Kurtosis:**	8.44

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

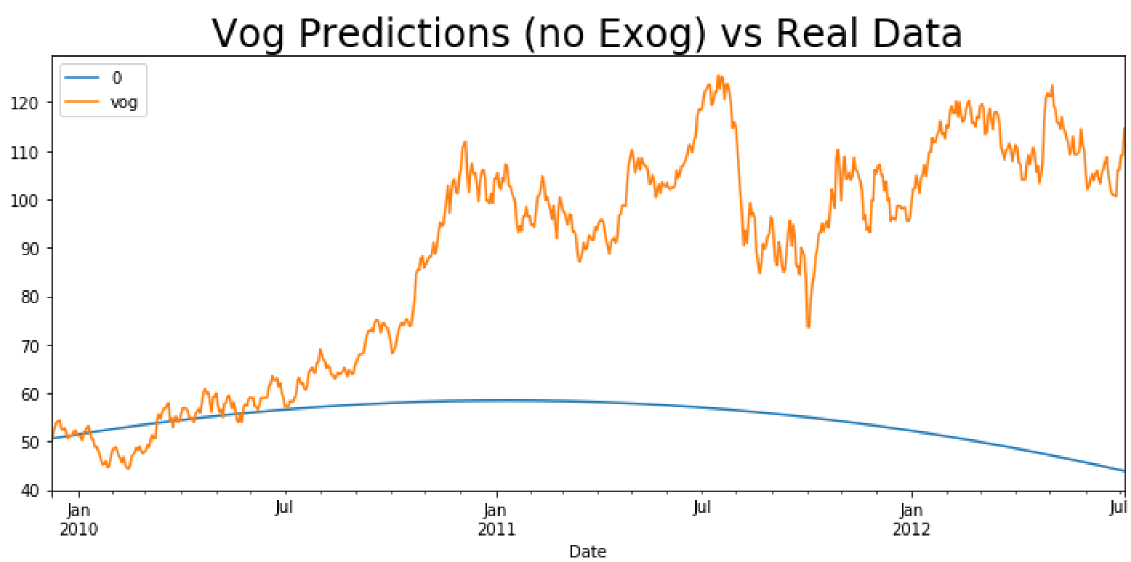
Future prediction of prices for Vog

In [96]:

```
#future prediction from start to the first announcement
model_vog_prediction = auto_arima(df.vog[start:f_ann], m = 5, max_p = 5, max_q = 5, max_P = 5,
max_Q = 5, trend = "ct")

prediction_vog = pd.DataFrame(model_vog_prediction.predict(n_periods = len(df[f_ann:s_ann])),
index = df[f_ann:s_ann].index)
prediction_vog[f_ann:s_ann].plot(figsize = (12,5), legend=True)

df.vog[f_ann:s_ann].plot(legend=True)
plt.title("Vog Predictions (no Exog) vs Real Data", size = 24)
```

Out[96]:

```
Text(0.5, 1.0, 'Vog Predictions (no Exog) vs Real Data')
```

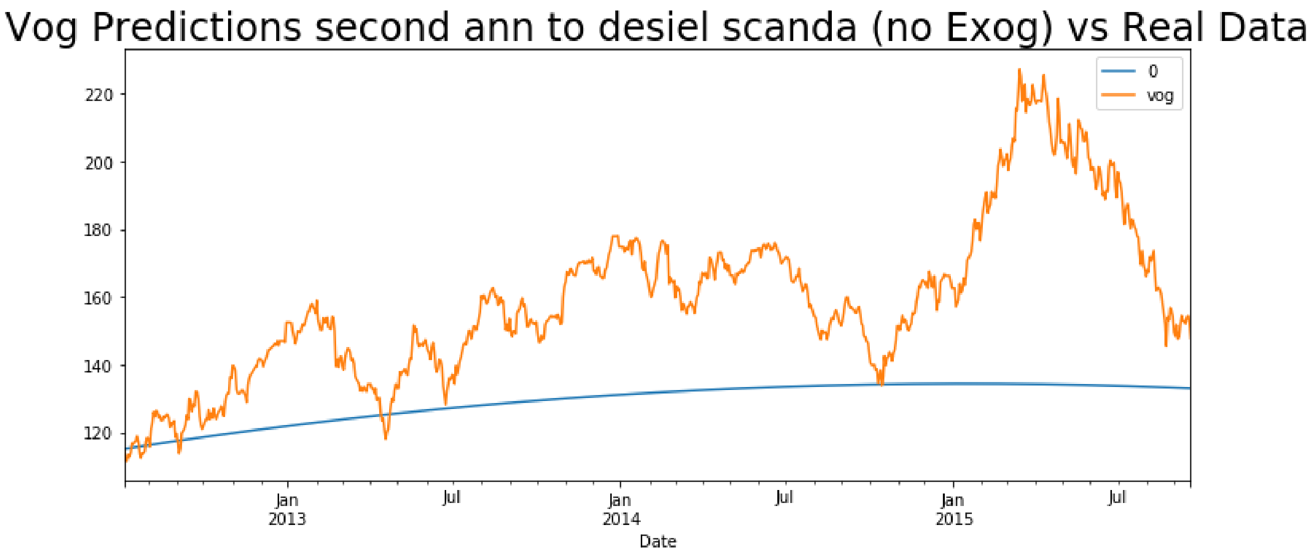


In [97]:

```
#future prediction from firts to the second announcement
model_vog_prediction = auto_arima(df.vog[f_ann:s_ann], m = 5, max_p = 5, max_q = 5, max_P = 5,
max_Q = 5, trend = "ct")

prediction_vog = pd.DataFrame(model_vog_prediction.predict(n_periods = len(df[s_ann:d_sca])),
index = df[s_ann:d_sca].index)
prediction_vog[s_ann:d_sca].plot(figsize = (12,5), legend=True)

df.vog[s_ann:d_sca].plot(legend=True)
plt.title("Vog Predictions second ann to desiel scanda (no Exog) vs Real Data", size = 24)
```

Out[97]:

```
Text(0.5, 1.0, 'Vog Predictions second ann to desiel scanda (no Exog) vs Real Data')
```

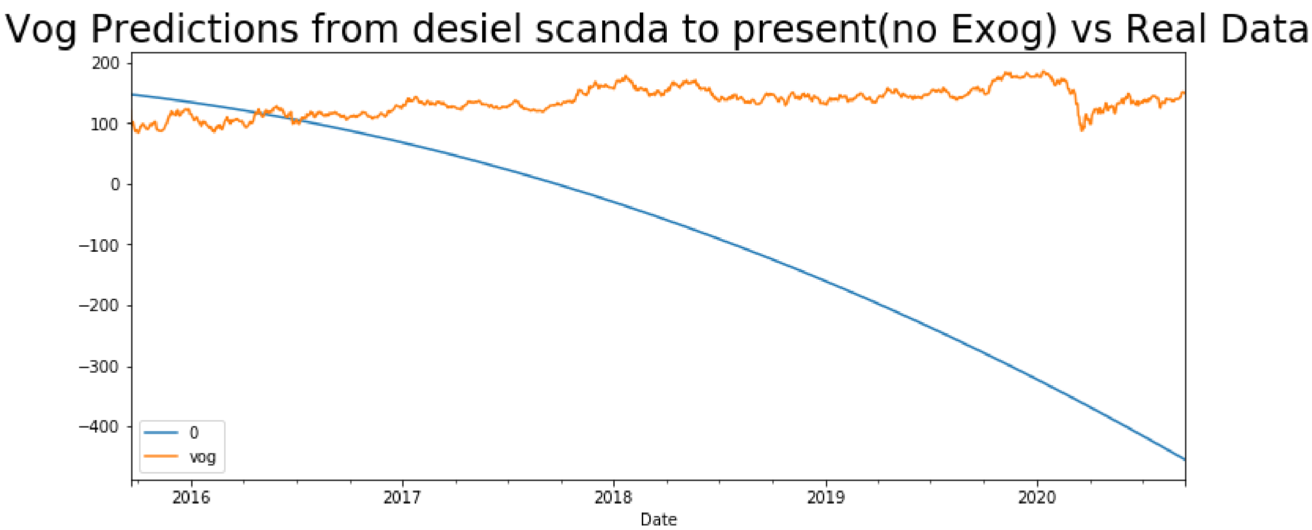


In [99]:

```
#future prediction from desiel scandal to present
model_vog_prediction = auto_arima(df.vog[s_ann:d_sca], m = 5, max_p = 5, max_q = 5, max_P = 5,
max_Q = 5, trend = "ct")

prediction_vog = pd.DataFrame(model_vog_prediction.predict(n_periods = len(df[d_sca:'2020-09-14']))
,
                              index = df[d_sca:'2020-09-14'].index)
prediction_vog[d_sca:'2020-09-14'].plot(figsize = (12,5), legend=True)

df.vog[d_sca:'2020-09-14'].plot(legend=True)
plt.title("Vog Predictions from desiel scanda to present(no Exog) vs Real Data", size = 24)
```

Out[99]:

```
Text(0.5, 1.0, 'Vog Predictions from desiel scanda to present(no Exog) vs Real Data')
```

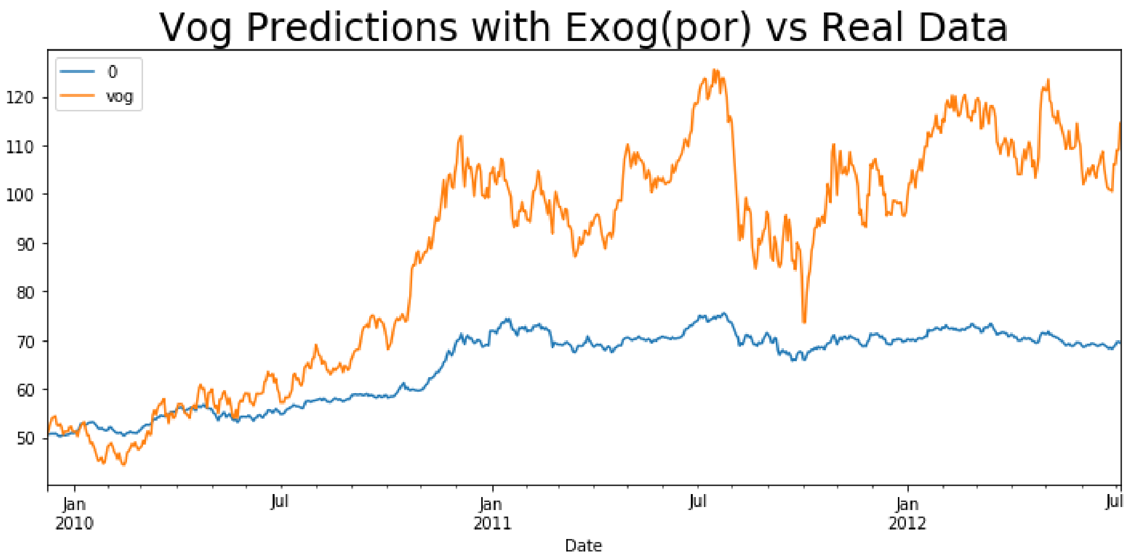


In [102]:

```
#future prediction from start to the first announcement with por as exogenous variable
model_vog_prediction = auto_arima(df.vog[start:f_ann], exogenous = df[['por']][start:f_ann] ,m = 5,
max_p = 5, max_q = 5, max_P = 5, max_Q = 5, trend = "ct")

prediction_vog_exP = pd.DataFrame(model_vog_prediction.predict(n_periods =
len(df[f_ann:s_ann]),exogenous = df[['por']][f_ann:s_ann]), index = df[f_ann:s_ann].index)
prediction_vog_exP[f_ann:s_ann].plot(figsize = (12,5), legend=True)

df.vog[f_ann:s_ann].plot(legend=True)
plt.title("Vog Predictions with Exog(por) vs Real Data", size = 24)
```

Out[102]:

```
Text(0.5, 1.0, 'Vog Predictions with Exog(por) vs Real Data')
```

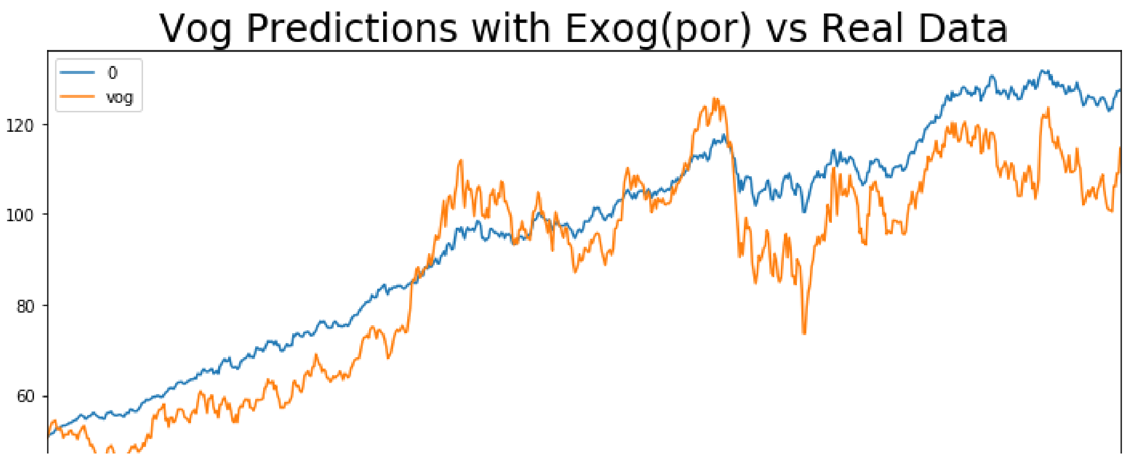


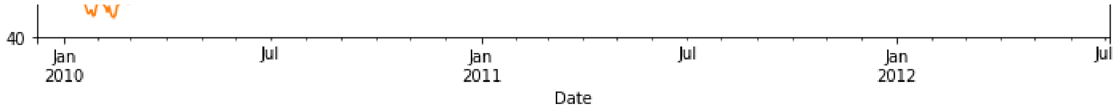
In [103]:

```
#future prediction from start to the first announcement with por as exogenous variable
model_vog_prediction = auto_arima(df.vog[start:f_ann], exogenous = df[['bmw']][start:f_ann] ,m = 5,
max_p = 5, max_q = 5, max_P = 5, max_Q = 5, trend = "ct")

prediction_vog_exB = pd.DataFrame(model_vog_prediction.predict(n_periods =
len(df[f_ann:s_ann]),exogenous = df[['bmw']][f_ann:s_ann]), index = df[f_ann:s_ann].index)
prediction_vog_exB[f_ann:s_ann].plot(figsize = (12,5), legend=True)

df.vog[f_ann:s_ann].plot(legend=True)
plt.title("Vog Predictions with Exog(bmw) vs Real Data", size = 24)
```

Out[103]:

```
Text(0.5, 1.0, 'Vog Predictions with Exog(por) vs Real Data')
```



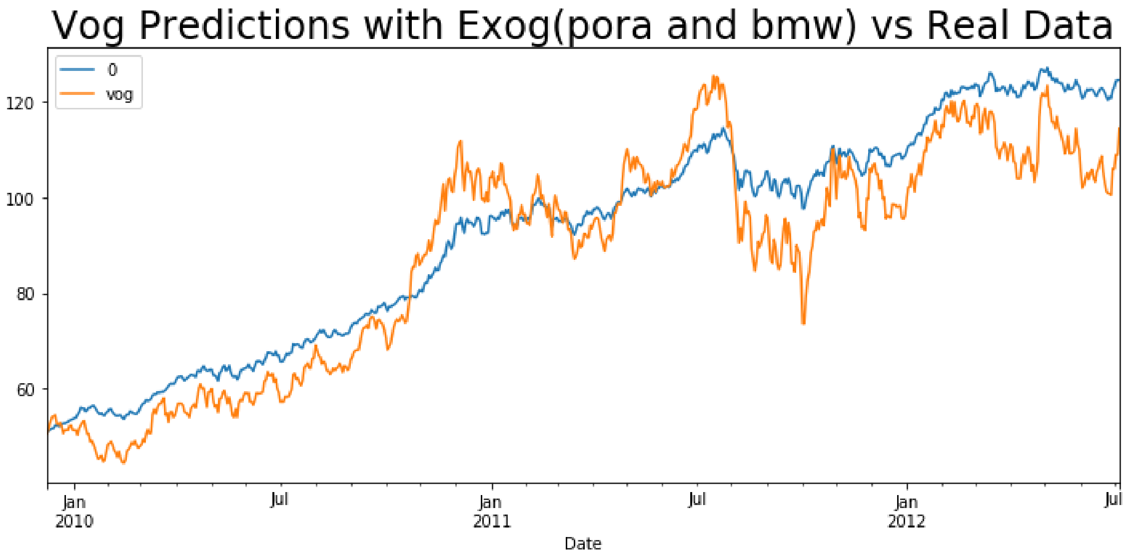


In [104]:

```
#future prediction from start to the first announcement with both por and bmw as exogenous variabl
e
model_vog_prediction = auto_arima(df.vog[start:f_ann], exogenous = df[['por','bmw']][start:f_ann] ,
m = 5, max_p = 5, max_q = 5, max_P = 5, max_Q = 5, trend = "ct")

prediction_vog_ex = pd.DataFrame(model_vog_prediction.predict(n_periods =
len(df[f_ann:s_ann]),exogenous = df[['por','bmw']][f_ann:s_ann]), index = df[f_ann:s_ann].index)
prediction_vog_ex[f_ann:s_ann].plot(figsize = (12,5), legend=True)

df.vog[f_ann:s_ann].plot(legend=True)
plt.title("Vog Predictions with Exog(pora and bmw) vs Real Data", size = 24)
```

Out[104]:

```
Text(0.5, 1.0, 'Vog Predictions with Exog(pora and bmw) vs Real Data')
```

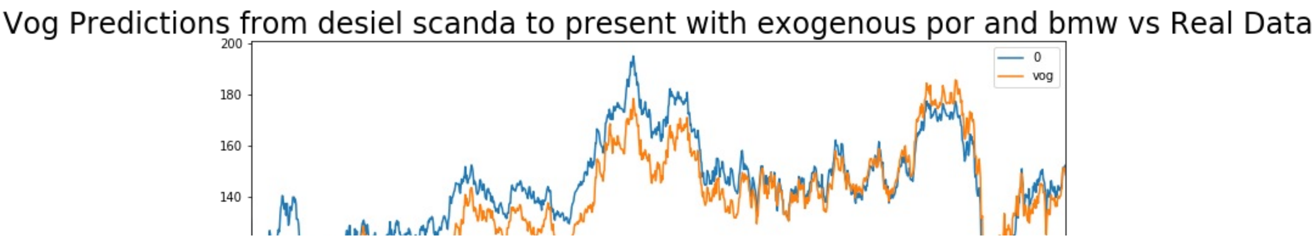


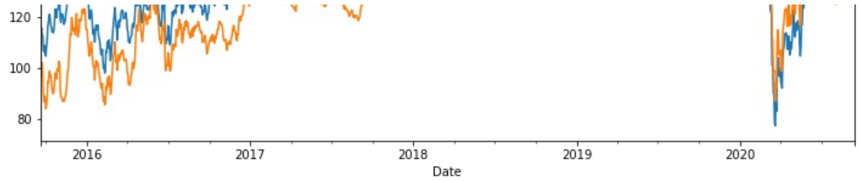
In [108]:

```
#future prediction from desiel scandal to present
model_vog_prediction = auto_arima(df.vog[s_ann:d_sca],exogenous = df[['por','bmw']][s_ann:d_sca],
m = 5, max_p = 5, max_q = 5, max_P = 5, max_Q = 5, trend = "ct")

prediction_vogPB = pd.DataFrame(model_vog_prediction.predict(n_periods = len(df[d_sca:'2020-09-14']
),exogenous = df[['por','bmw']][d_sca:'2020-09-14']),
                              index = df[d_sca:'2020-09-14'].index)
prediction_vogPB[d_sca:'2020-09-14'].plot(figsize = (12,5), legend=True)

df.vog[d_sca:'2020-09-14'].plot(legend=True)
plt.title("Vog Predictions from desiel scandal to present with exogenous por and bmw vs Real Data"
, size = 24)
```

Out[108]:

```
Text(0.5, 1.0, 'Vog Predictions from desiel scanda to present with exogenous por and bmw vs Real D
ata')
```

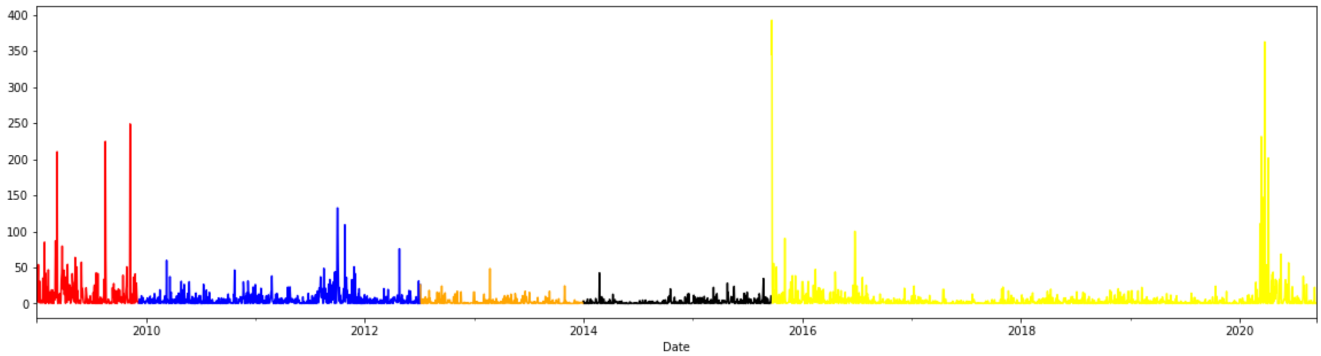




Lets examine the volatility of Vol from start date to desiel scandal

In [110]:

```
df['sq_vog'][start:f_ann].plot(figsize = (20,5), color = 'red')
df['sq_vog'][f_ann:s_ann].plot(color = 'blue')
df['sq_vog'][s_ann:end].plot(color = 'orange')
df['sq_vog'][end:d_sca].plot(color = 'black')
df['sq_vog'][d_sca:'2020-09-14'].plot(color = 'yellow')
plt.show()
```



In [112]:

```
#Volatility trend for each period
from arch import arch_model
```

In [115]:

```
model_garch_start_f = arch_model(df.ret_vog[start:f_ann], mean = "Constant",  vol = "GARCH", p = 1,
q = 1)
results_garch_start_f = model_garch_start_f.fit(update_freq = 5)

model_garch_f_s = arch_model(df.ret_vog[f_ann:s_ann], mean = "Constant",  vol = "GARCH", p = 1, q =
1)
results_garch_f_s = model_garch_f_s.fit(update_freq = 5)

model_garch_s_e = arch_model(df.ret_vog[s_ann:end], mean = "Constant",  vol = "GARCH", p = 1, q = 1
)
results_garch_s_e = model_garch_s_e.fit(update_freq = 5)

model_garch_e_dc = arch_model(df.ret_vog[end:d_sca], mean = "Constant",  vol = "GARCH", p = 1, q =
1)
results_garch_e_dc = model_garch_e_dc.fit(update_freq = 5)

model_garch_dc_pre = arch_model(df.ret_vog[d_sca:'2020-09-14'], mean = "Constant",  vol = "GARCH",
p = 1, q = 1)
results_garch_dc_pre = model_garch_dc_pre.fit(update_freq = 5)
```

```
Iteration:      5,   Func. Count:     35,   Neg. LLF: 673.2438089672729
Iteration:     10,   Func. Count:     65,   Neg. LLF: 673.2402452231768
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 673.2401928099462
            Iterations: 12
            Function evaluations: 77
            Gradient evaluations: 12
Iteration:      5,   Func. Count:     42,   Neg. LLF: 1526.9228402592855
Iteration:     10,   Func. Count:     72,   Neg. LLF: 1526.7316954368935
Optimization terminated successfully.    (Exit mode 0)
```

```
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 1526.7316954356486
            Iterations: 10
            Function evaluations: 72
            Gradient evaluations: 10
Iteration:      5,   Func. Count:     38,   Neg. LLF: 724.5620364427224
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 724.5578157320635
            Iterations: 7
            Function evaluations: 51
            Gradient evaluations: 7
Iteration:      5,   Func. Count:     40,   Neg. LLF: 825.496351472824
Iteration:     10,   Func. Count:     73,   Neg. LLF: 825.4707320291793
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 825.4707320282179
            Iterations: 10
            Function evaluations: 73
            Gradient evaluations: 10
Iteration:      5,   Func. Count:     38,   Neg. LLF: 2713.6026698362484
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 2713.5708492110825
            Iterations: 9
            Function evaluations: 63
            Gradient evaluations: 9
```

In [116]:

```
results_garch_start_f.summary()
```

Out[116]:

Constant Mean - GARCH Model Results

Dep. Variable:	ret_vog	**R-squared:**	-0.000
Mean Model:	Constant Mean	**Adj. R-squared:**	-0.000
Vol Model:	GARCH	**Log-Likelihood:**	-673.240
Distribution:	Normal	**AIC:**	1354.48
Method:	Maximum Likelihood	**BIC:**	1368.49
		No. Observations:	245
Date:	Wed, Sep 16 2020	**Df Residuals:**	241
Time:	20:40:31	**Df Model:**	4

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	0.2661	0.283	0.940	0.347	[-0.289, 0.821]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	6.2783	9.790	0.641	0.521	[-12.910, 25.466]
alpha[1]	0.0372	0.115	0.323	0.747	[-0.189, 0.263]
beta[1]	0.5200	0.755	0.688	0.491	[-0.961, 2.001]

Covariance estimator: robust

In [117]:

```
results_garch_f_s.summary()
```

Out[117]:

Constant Mean - GARCH Model Results

Dep. Variable:	ret_vog	**R-squared:**	-0.000
Mean Model:	Constant Mean	**Adj. R-squared:**	-0.000

Mean Model:	Constant Mean	**Adj. R-squared:**	-0.000
Vol Model:	GARCH	**Log-Likelihood:**	-1526.73
Distribution:	Normal	**AIC:**	3061.46
Method:	Maximum Likelihood	**BIC:**	3079.50
		No. Observations:	672
Date:	Wed, Sep 16 2020	**Df Residuals:**	668
Time:	20:40:31	**Df Model:**	4

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	0.1892	8.634e-02	2.191	2.843e-02	[1.998e-02, 0.358]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	0.1679	7.914e-02	2.122	3.388e-02	[1.279e-02, 0.323]
alpha[1]	0.0688	1.682e-02	4.091	4.301e-05	[3.585e-02, 0.102]
beta[1]	0.9040	2.108e-02	42.883	0.000	[0.863, 0.945]

Covariance estimator: robust

In [118]:

```
results_garch_s_e.summary()
```

Out[118]:

Constant Mean - GARCH Model Results

Dep. Variable:	ret_vog	**R-squared:**	-0.001
Mean Model:	Constant Mean	**Adj. R-squared:**	-0.001
Vol Model:	GARCH	**Log-Likelihood:**	-724.558
Distribution:	Normal	**AIC:**	1457.12
Method:	Maximum Likelihood	**BIC:**	1472.98
		No. Observations:	390
Date:	Wed, Sep 16 2020	**Df Residuals:**	386
Time:	20:40:31	**Df Model:**	4

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	0.2298	9.845e-02	2.334	1.958e-02	[3.685e-02, 0.423]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	0.7719	0.677	1.141	0.254	[-0.554, 2.098]
alpha[1]	0.1853	0.145	1.273	0.203	[-9.991e-02, 0.470]
beta[1]	0.5136	0.331	1.550	0.121	[-0.136, 1.163]

Covariance estimator: robust

In [119]:

```
results_garch_e_dc.summary()
```

Out[119]:

Constant Mean - GARCH Model Results

Dep. Variable:	ret_vog	**R-squared:**	-0.000
Mean Model:	Constant Mean	**Adj. R-squared:**	-0.000
Vol Model:	GARCH	**Log-Likelihood:**	-825.471
Distribution:	Normal	**AIC:**	1658.94
Method:	Maximum Likelihood	**BIC:**	1675.36
		No. Observations:	448
Date:	Wed, Sep 16 2020	**Df Residuals:**	444
Time:	20:40:31	**Df Model:**	4

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	-0.0470	6.969e-02	-0.675	0.500	[-0.184,8.957e-02]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	0.0213	2.156e-02	0.987	0.323	[-2.097e-02,6.356e-02]
alpha[1]	0.0479	2.574e-02	1.859	6.305e-02	[-2.603e-03,9.831e-02]
beta[1]	0.9466	2.803e-02	33.776	4.480e-250	[0.892, 1.001]

Covariance estimator: robust

In [120]:

```
results_garch_dc_pre.summary()
```

Out[120]:

Constant Mean - GARCH Model Results

Dep. Variable:	ret_vog	**R-squared:**	-0.001
Mean Model:	Constant Mean	**Adj. R-squared:**	-0.001
Vol Model:	GARCH	**Log-Likelihood:**	-2713.57
Distribution:	Normal	**AIC:**	5435.14
Method:	Maximum Likelihood	**BIC:**	5455.83
		No. Observations:	1301
Date:	Wed, Sep 16 2020	**Df Residuals:**	1297
Time:	20:40:31	**Df Model:**	4

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	0.0929	4.845e-02	1.918	5.512e-02	[-2.035e-03, 0.188]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	0.0905	3.859e-02	2.346	1.895e-02	[1.491e-02, 0.166]
alpha[1]	0.0868	2.522e-02	3.441	5.804e-04	[3.734e-02, 0.136]
beta[1]	0.8907	2.787e-02	31.958	4.165e-224	[0.836, 0.945]

Covariance estimator: robust

In [157]:

```
model_garch_final = arch_model(df.ret_vog[start:'2020-09-14'], mean = "Constant",  vol = "GARCH", p
= 1, q = 1)
results_garch_final = model_garch_final.fit(last_obs=d_sca,update_freq=5)
```

```
Iteration:      5,   Func. Count:     36,   Neg. LLF: 3772.697091936414
Iteration:     10,   Func. Count:     71,   Neg. LLF: 3769.960813970977
Iteration:     15,   Func. Count:    102,   Neg. LLF: 3769.9520034944235
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 3769.9520034942498
            Iterations: 15
            Function evaluations: 102
            Gradient evaluations: 15
```

In [158]:

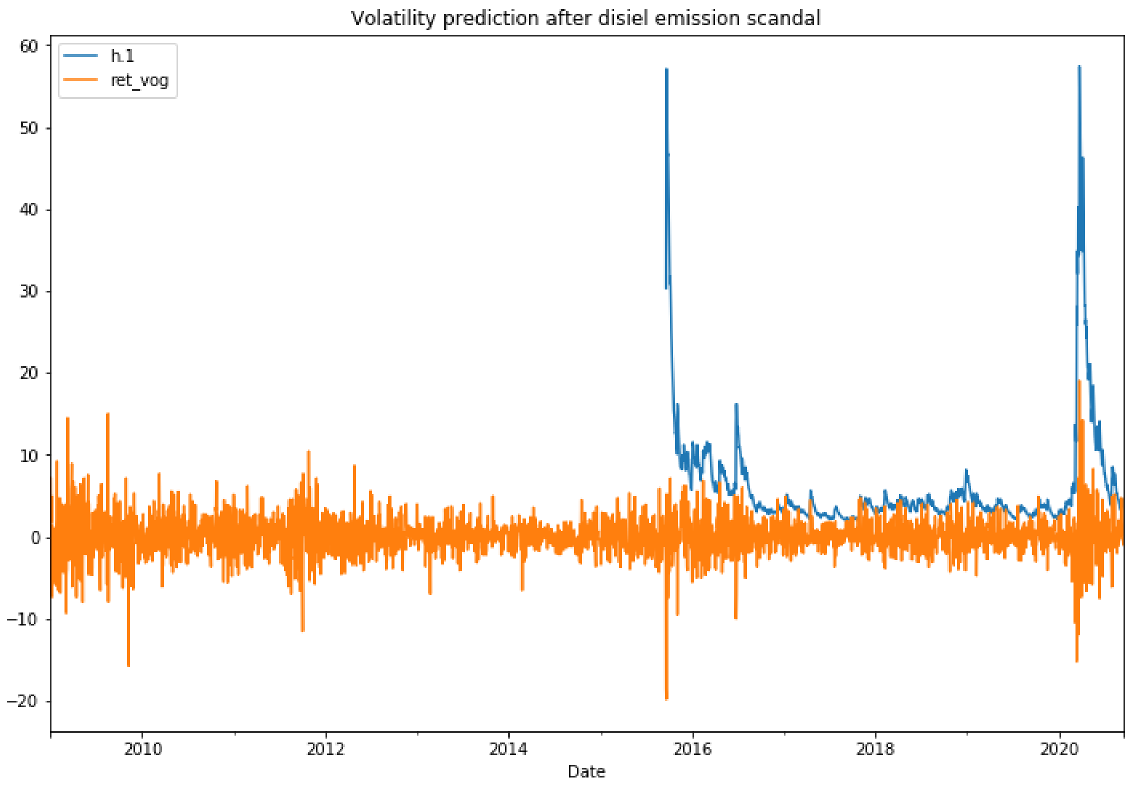
```
forecast_garch=results_garch_final.forecast()
```

In [164]:

```
forecast_garch.residual_variance[d_sca:'2020-09-14'].plot(figsize=(12,8),legend=True)
df.ret_vog[start:'2020-09-14'].plot(legend=True)
plt.title('Volatility prediction after diesel emission scandal')
```

Out[164]:

```
Text(0.5, 1.0, 'Volatility prediction after disiel emission scandal')
```



In [160]:

```
model_garch_final = arch_model(df.ret_vog[start:'2020-09-14'], mean = "Constant",  vol = "GARCH", p
= 1, q = 1)
results_garch_final = model_garch_final.fit(last_obs=s_ann,update_freq=5)
```

```
Iteration:      5,   Func. Count:     38,   Neg. LLF: 2207.012544659883
Iteration:     10,   Func. Count:     71,   Neg. LLF: 2206.690006449945
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 2206.690003959738
            Iterations: 11
            Function evaluations: 77
            Gradient evaluations: 11
```

In [161]:

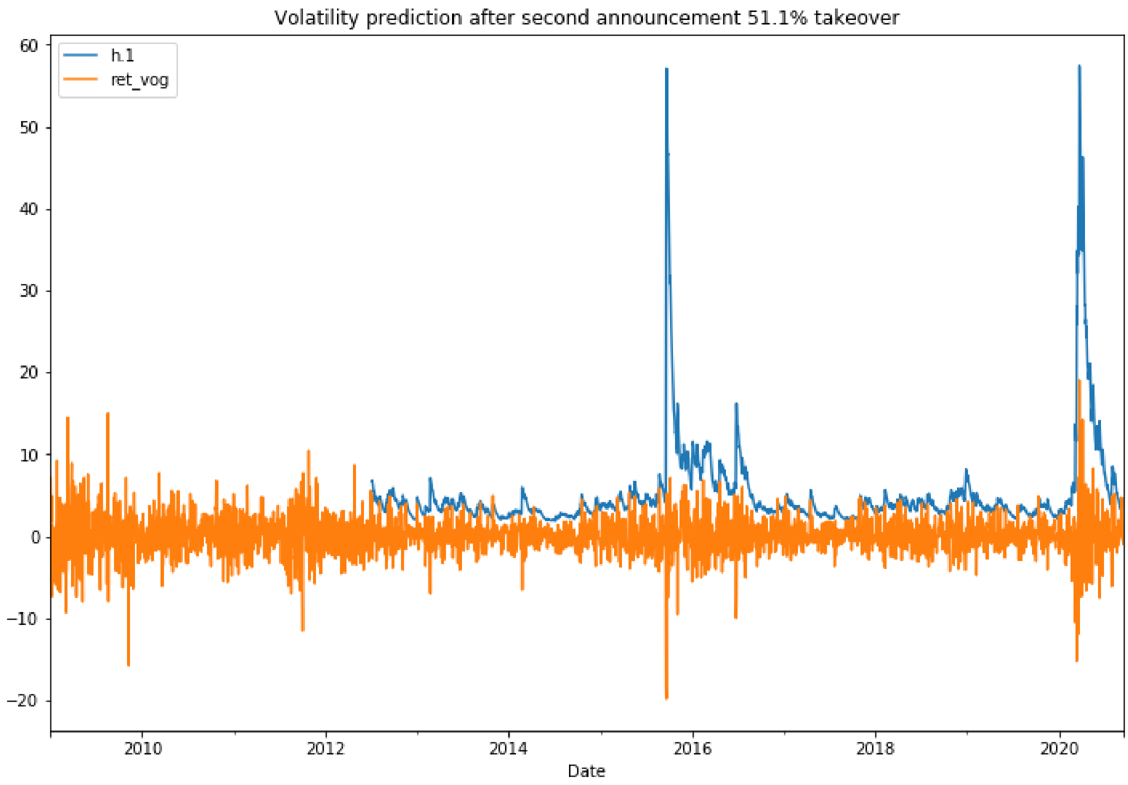
```
forecast_garch=results_garch_final.forecast()
```

In [165]:

```
forecast_garch.residual_variance[s_ann:'2020-09-14'].plot(figsize=(12,8),legend=True)
df.ret_vog[start:'2020-09-14'].plot(legend=True)
plt.title('Volatility prediction after second announcement 51.1% takeover')
```

Out[165]:

```
Text(0.5, 1.0, 'Volatility prediction after second announcement 51.1% takeover')
```



In [ ]: