

Práctica 3: BMW VS VOLKSWAGEN

Guillermo González Díaz

Noviembre de 2021

Preguntas a responder:

Utilizando datos diarios de yahoo de los dos series (yfinance), debemos analizar la relación entre BMW (BMW.DE) y Volkswagen(VOW.DE), contestando a las siguiente preguntas:

1. Estima un modelo VAR mensual entre los dos activos. **(Muestra:2009/01 a 2020/10)**
2. ¿Cuál es la predicción para octubre de 2021 de los dos activos?. ¿Son buenas predicciones?
3. ¿Cuál de los dos activos es el lider y cual es el seguidor? Cual explica a cual vaya
4. ¿Es la Volatilidad de BMW constante? Estimar el modelo garch de BMW (Muestra: Toda la disponible en Yahoo hasta el 31/10/2020)
5. ¿Cuál es la predicción para el mes de noviembre de 2020? ¿Son buenas predicciones?
6. Plantea un sistema de trading diario para BMW apartir del modelo Garch para el mes de noviembre de 2020. ¿Es rentable? Esto quiere decir que con la volatilidad plantear modelos de inversión

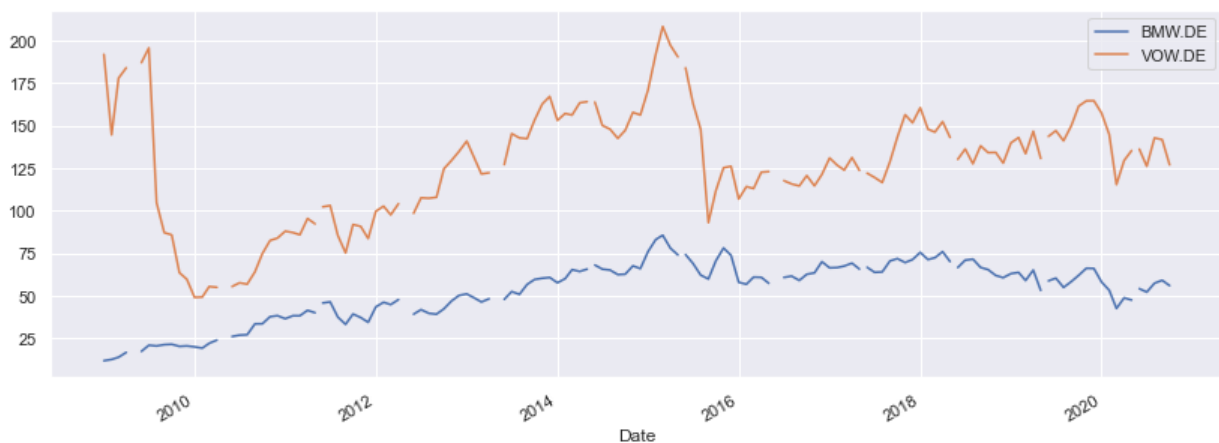
1. Carga y visualización de nuestros datos:

En este caso solo nos interesan los datos de cierre ajustado, por eso creamos un Data Frame con las dos series en el período comprendido entre enero de 2009 y octubre de 2020, ambos incluidos.

Nos quedamos solamente don el *precio ajustado*, que es el que nos interesa

Visualizamos nuestra serie temporal conjunta y comentamos Volkswagen aparte. Podemos observar que las dos llevan una tendencia parecida, creciendo después de la crisisi de 2009 hasta el momento del escándalo de las emisiones, y después una recuperación constante hasta la siguiente gran caída, el año pasado, causada por el covid.

```
<AxesSubplot:xlabel='Date'>
```



Mirando Volkswagen para ver mejor los tres momentos críticos de los que hablábamos antes. Es especialmente llamativa la caída de 2009 y la provocada por la estafa de las emisiones, mucho más drásticas que la del año pasado. Aquí el gráfico:

Calculamos los retornos logarítmicos y visualizamos nuevamente las series:

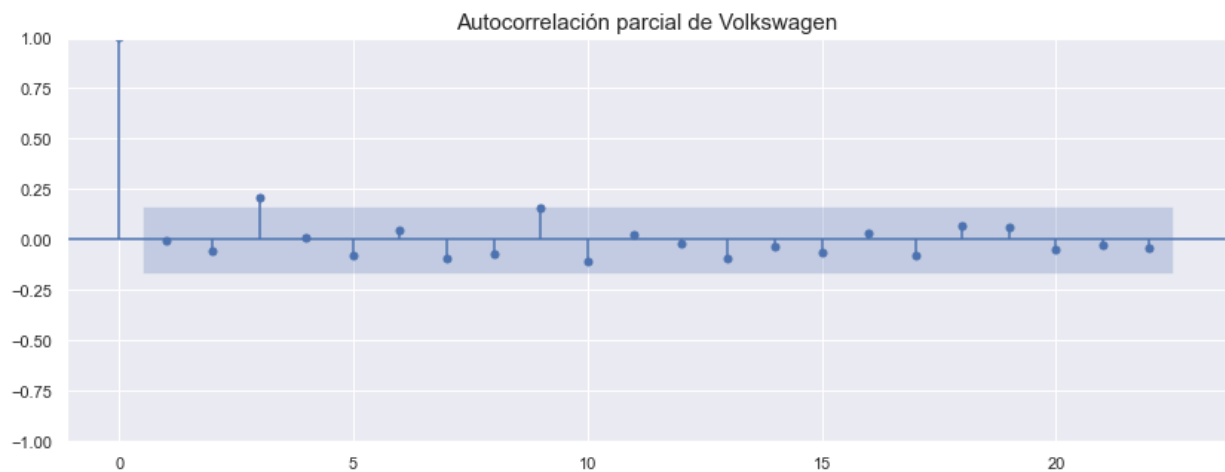
<AxesSubplot:xlabel='Date'>



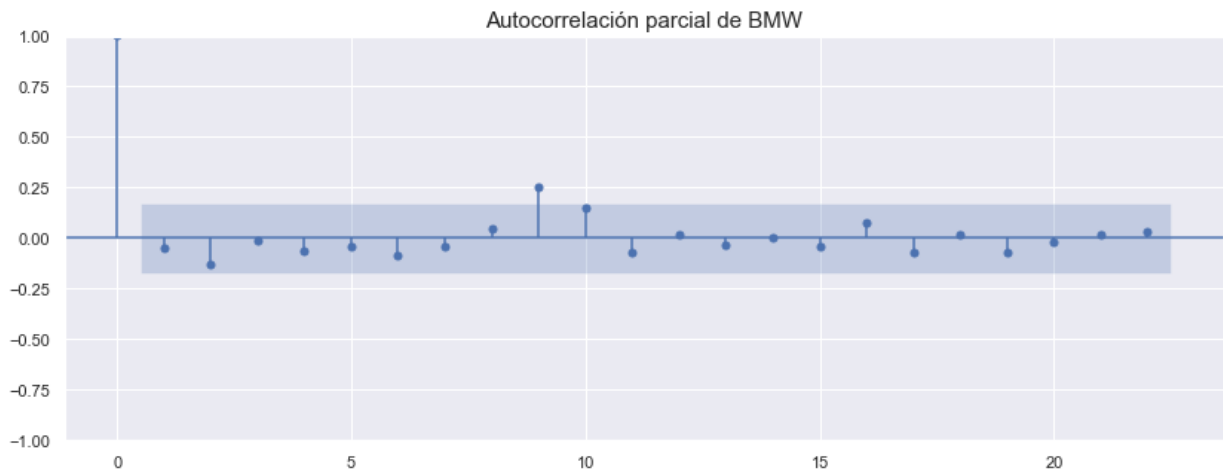
Autocorrelación:

A continuación pasamos a visualizar si existe autocorrelación parcial en nuestras dos series. En las dos solamente existe un período que sea significativo y pueda explicar el actual, por lo tanto podemos decir que no existe autocorrelación en ambas series. Vemos que estas dos series son estacionarias en sus retornos, por eso a continuación pasaremos a realizar un modelo VAR

```
/Users/guillemrogonzalezdiaz/miniconda3/lib/python3.8/site-packages/statsmodels/graphics/tsaplots.py:348: FutureWarning: The default method 'yw' can produce PACF values outside of the [-1,1] interval. After 0.13, the default will change to unadjusted Yule-Walker ('ywm'). You can use this method now by setting method='ywm'.
warnings.warn(
```



```
/Users/guillemrogonzalezdiaz/miniconda3/lib/python3.8/site-packages/statsmodels/graphics/tsaplots.py:348: FutureWarning: The default method 'yw' can produce PACF values outside of the [-1,1] interval. After 0.13, the default will change to unadjusted Yule-Walker ('ywm'). You can use this method now by setting method='ywm'.
warnings.warn(
```



2. Causalidad

¿ Quién sigue a quién ?

Intentando contestar a la pregunta que se nos realiza, analizamos los siguientes resultados. Hemos realizado un test de Granger, que nos dice si una serie temporal viene causada por otra. En la línea de código, la que introducimos primero es la que explica a la que introducimos en segundo lugar. Tenemos que fijarnos en el p valor, si nos encontráramos con un p valor muy bajo, podríamos decir que la primera serie explica, o provoca algún efecto en la segunda.

Primero comentamos a BMW, que parece no explicar a Volkswagen en ninguno de los períodos. Hemos puesto 13 lags para ver un año entero más un mes, ya que enero-enero(-1) podrían estar relacionados.

Después, viendo Volkswagen como líder vemos que tampoco existe causalidad. He probado con hasta 13 retardos para comparar poco más de un año y en ningún momento he detectado esa correlación, por lo tanto dejo un lag de 3 para que sirva como ejemplo. Visualizamos los resultados del primero:

Test de causalidad de Granger:

```

Granger Causality
number of lags (no zero) 1
ssr based F test:      F=2.3004 , p=0.1320 , df_denom=120, df_num=1
ssr based chi2 test:   chi2=2.3579 , p=0.1246 , df=1
likelihood ratio test: chi2=2.3356 , p=0.1264 , df=1
parameter F test:      F=2.3004 , p=0.1320 , df_denom=120, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:      F=1.1502 , p=0.3201 , df_denom=117, df_num=2
ssr based chi2 test:   chi2=2.3986 , p=0.3014 , df=2
likelihood ratio test: chi2=2.3754 , p=0.3049 , df=2
parameter F test:      F=1.1502 , p=0.3201 , df_denom=117, df_num=2

Granger Causality
number of lags (no zero) 3
ssr based F test:      F=0.8398 , p=0.4748 , df_denom=114, df_num=3
ssr based chi2 test:   chi2=2.6741 , p=0.4446 , df=3
likelihood ratio test: chi2=2.6450 , p=0.4497 , df=3
parameter F test:      F=0.8398 , p=0.4748 , df_denom=114, df_num=3

```

3. Modelo VAR

Realizamos una selección de VAR automática y vemos que nos arroja un modelo con dos ecuaciones. Hemos vuelto a usar el máximo de lags para que tenga todo ese período en cuenta.

Este sería un ejemplo de VAR 1 model porque sólo nos interesa un momento del pasado, si queremos más períodos las ecuaciones se complican más. Usamos vectores para estas fórmulas.

Hacemos el VAR sobre los retornos puesto que los precios no son estacionarios. Este es nuestro modelo:

```

Summary of Regression Results
=====
Model:                VAR
Method:               OLS
Date:                Thu, 25, Nov, 2021
Time:                15:38:59

-----
No. of Equations:    2.00000    BIC:                -9.70721
Nobs:                140.000    HQIC:               -9.78205
Log likelihood:      297.027    FPE:                5.36372e-05
AIC:                 -9.83328    Det(Omega_mle):     5.14103e-05

-----
Results for equation bmw_ret
=====
              coefficient      std. error      t-stat      prob
-----
const          0.009969        0.007129        1.398        0.162
L1.bmw_ret      0.040624        0.104033        0.390        0.696
L1.vk_ret      -0.096545        0.077948       -1.239        0.216

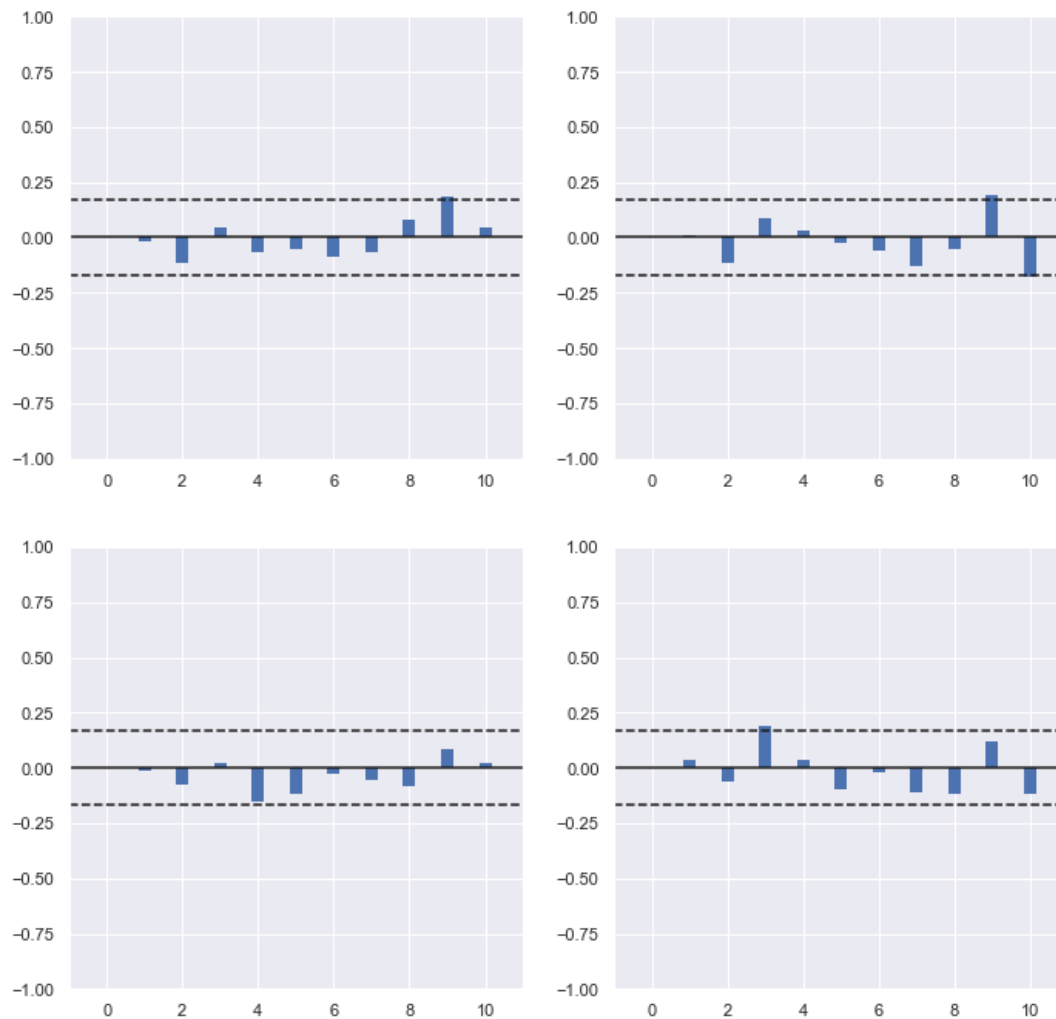
-----
Results for equation vk_ret
=====
              coefficient      std. error      t-stat      prob
-----
const          -0.000377        0.009367       -0.040        0.968
L1.bmw_ret     -0.044966        0.136705       -0.329        0.742
L1.vk_ret       0.016344        0.102428        0.160        0.873

-----
Correlation matrix of residuals
      bmw_ret    vk_ret
bmw_ret    1.000000    0.608948
vk_ret     0.608948    1.000000

```

Residuos Comprobamos que los residuos de nuestro modelo se comportan como ruido blanco

ACF plots for residuals with $2/\sqrt{T}$ bounds



Predicciones:

Realizamos las predicciones para octubre de 2020

Predicción 12 periodos VOLKSWAGEN:

Predicción 12 periodos BMW

Comparamos nuestras predicciones con los valores reales:

Comparando ambas tablas, podemos ver que son predicciones bastante malas. Si bien es cierto que para BMW no tanto como para Volkswagen, podemos decir que no es un buen modelo y que no se ajusta bien.

Predicción Octubre 2021

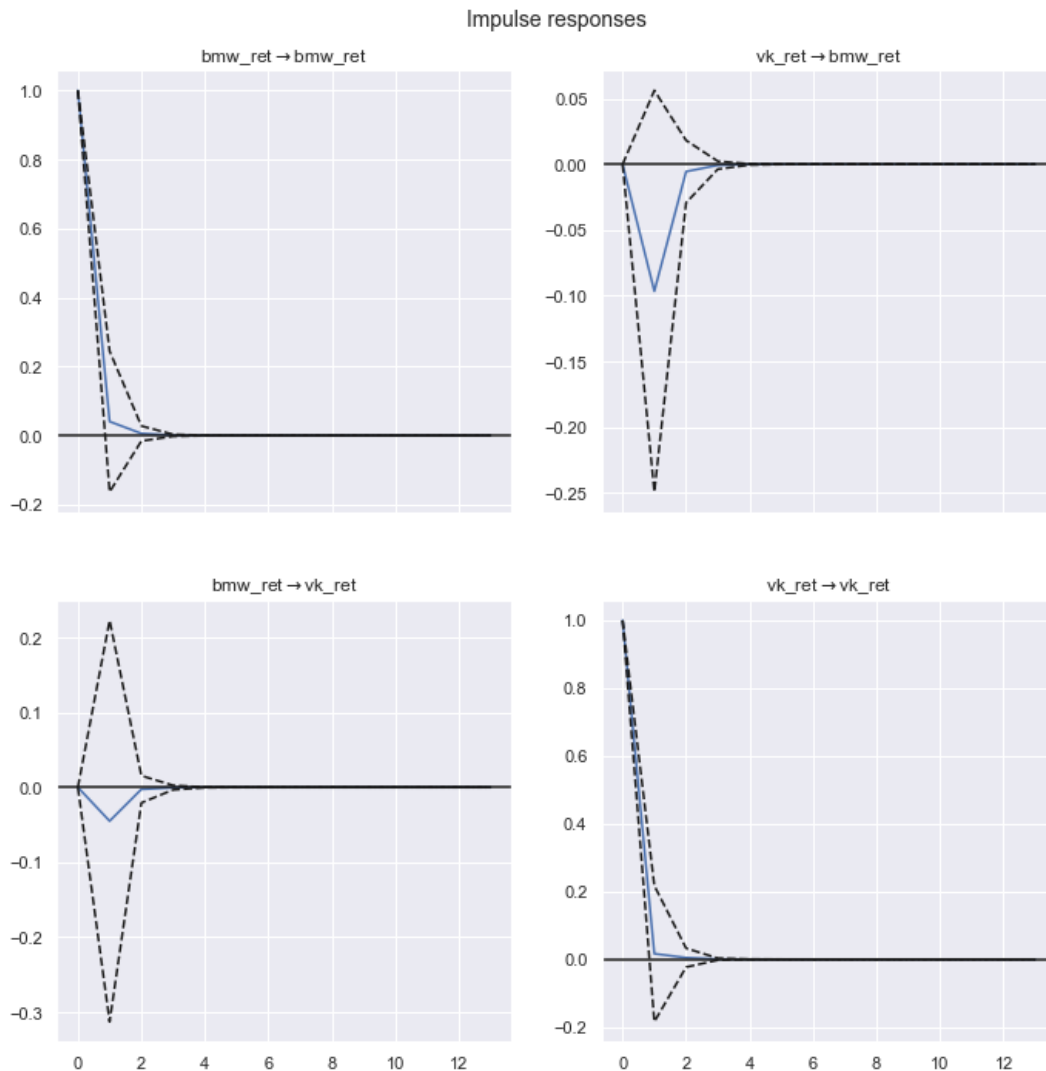
BMW.DE 64.731

VOW.DE 125.765

Valores reales de octubre de 2021:

Respuesta del impulso:

Observamos que no provocan efectos mutuos:



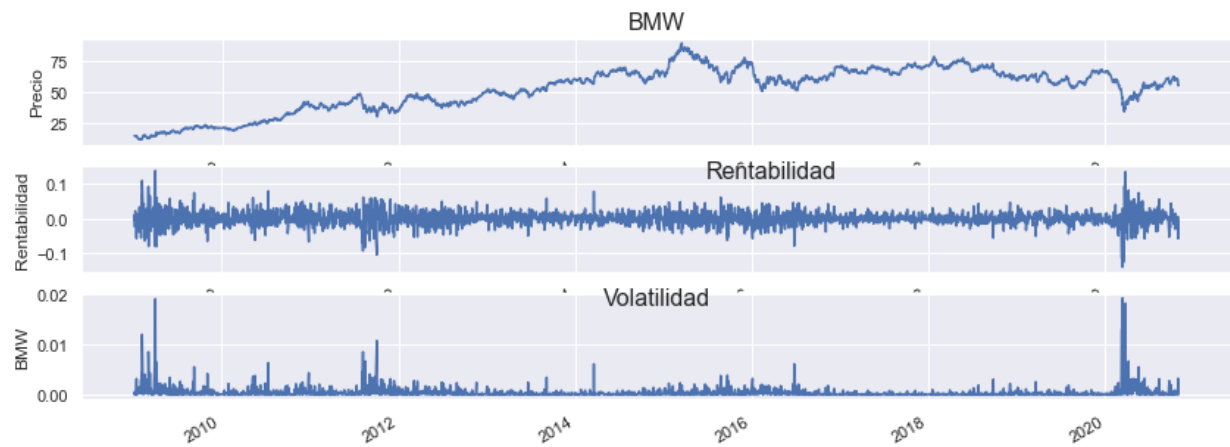
4. Volatilidad de BMW

En este apartado pasamos a analizar la volatilidad de bmw y a la realización de un modelo GARCH. Cogemos para ello los datos diarios del mismo período y los representamos:

A continuación, calculamos los rendimientos con la tasa de variación logarítmica. Los representamos. Después calculamos los rendimientos al cuadrado, que son una aproximación a la volatilidad.

Representación gráfica conjunta de precio, rentabilidad logarítmica y volatilidad:

```
Text(0.5, 0.8, 'Volatilidad')
```



¿Hay autocorrelación?

Podemos comprobar que sí existe correlación en la volatilidad de BMW, por lo tanto la z al cuadrado sigue teniendo comportamiento Garch, así que este modelo no es adecuado para predecir.

```
(<Figure size 864x576 with 3 Axes>,
 array([<AxesSubplot: >, <AxesSubplot: title='center': 'Autocorrelation'>,
        <AxesSubplot: title='center': 'Partial Autocorrelation'>],
 dtype=object))
```



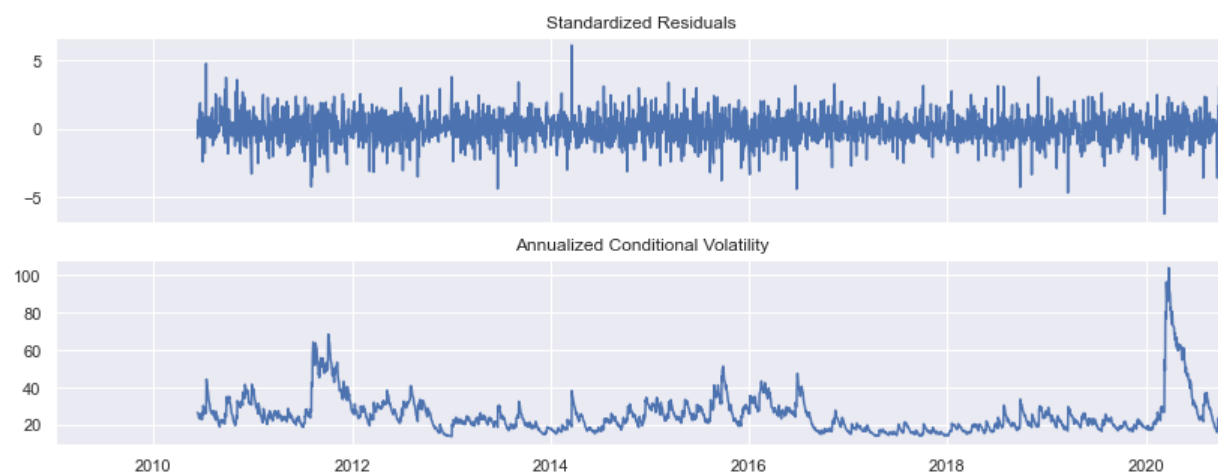
Podemos ver que existe mucha autocorrelación en la volatilidad, pero de momento cogeremos un Garch 2, 2, puesto que a partir del segundo lag de alguna forma se ve que empieza a caer. Hay más momentos en los que

existe autocorrelación, pero para mantenerlo siempre cogemos esos 2.

(Veremos que al final para este trabajo nos decantamos por un Garch 1, 1, ya que al ver los resultados del 2,2 tuve que cambiarlo)

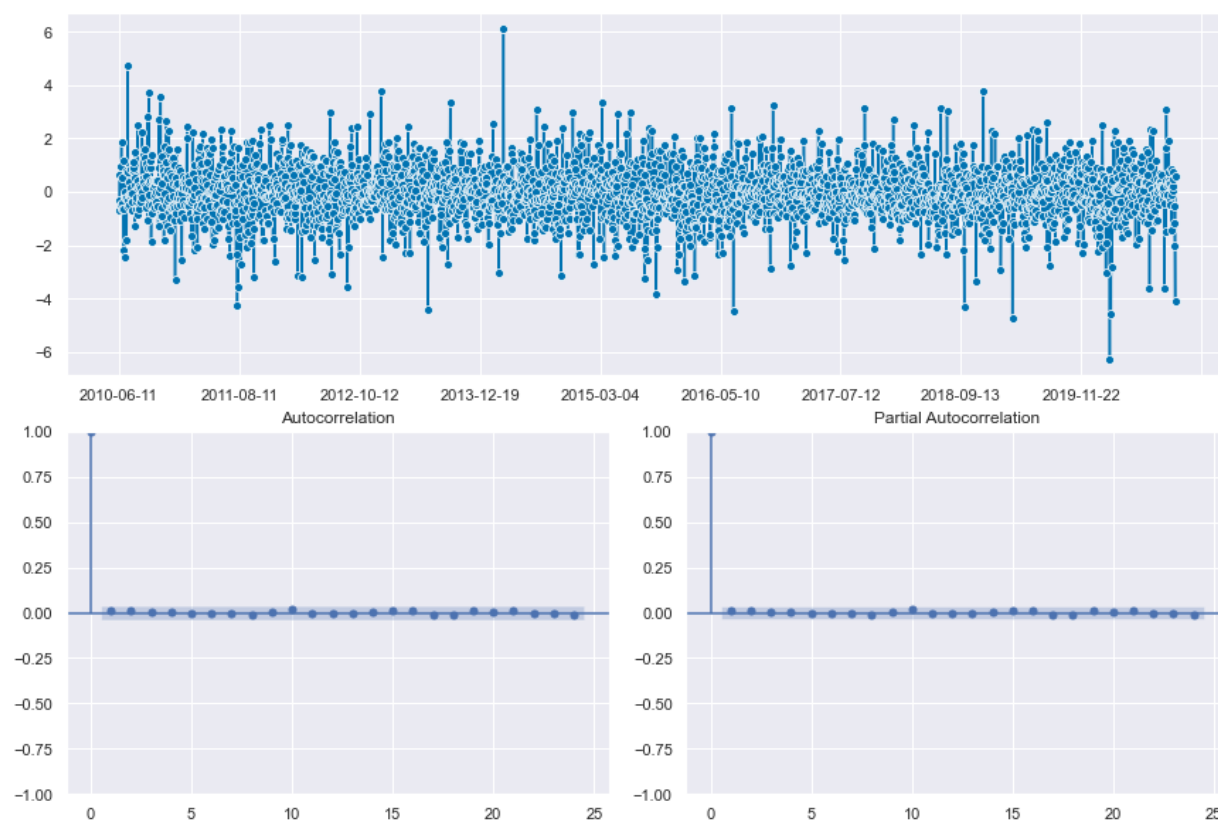
5. GARCH (1,1) BMW

Visualizamos los resultados anualizados: Podemos comprobar que esos clústeres que veíamos antes en la volatilidad de bmw se ha reducido bastante

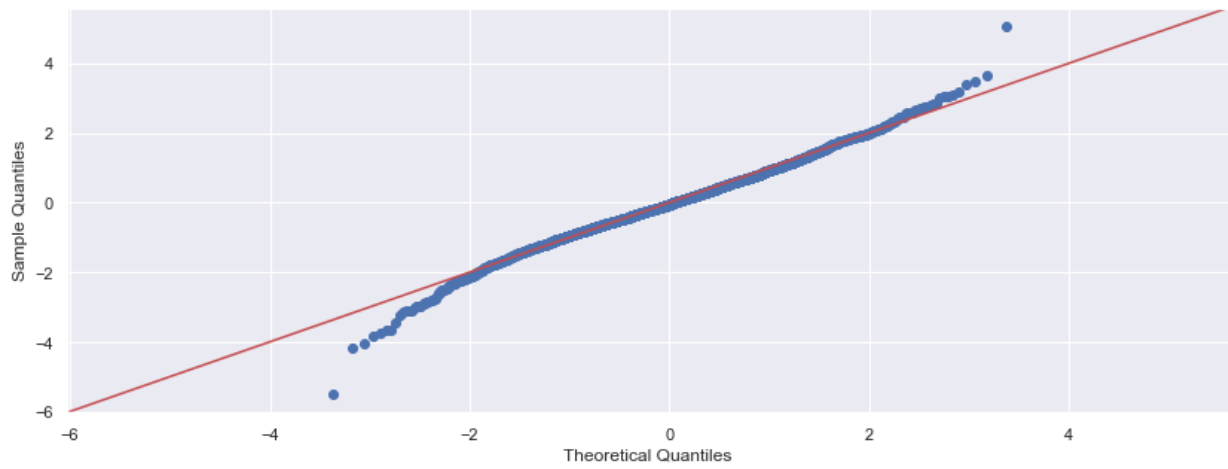


Analizamos los residuos: Igualmente ahora si existe ruido blanco en los residuos de BMW, no hay efecto GARCH.

```
(<Figure size 864x576 with 3 Axes>,  
array([<AxesSubplot: >, <AxesSubplot: title={ 'center': 'Autocorrelation' }>,  
       <AxesSubplot: title={ 'center': 'Partial Autocorrelation' }>],  
      dtype=object))
```



Vamos a ver como se distribuyen los residuos. Vemos que se ajustan bastante a la recta, pero vamos a probar con T Student a ver si mejora.

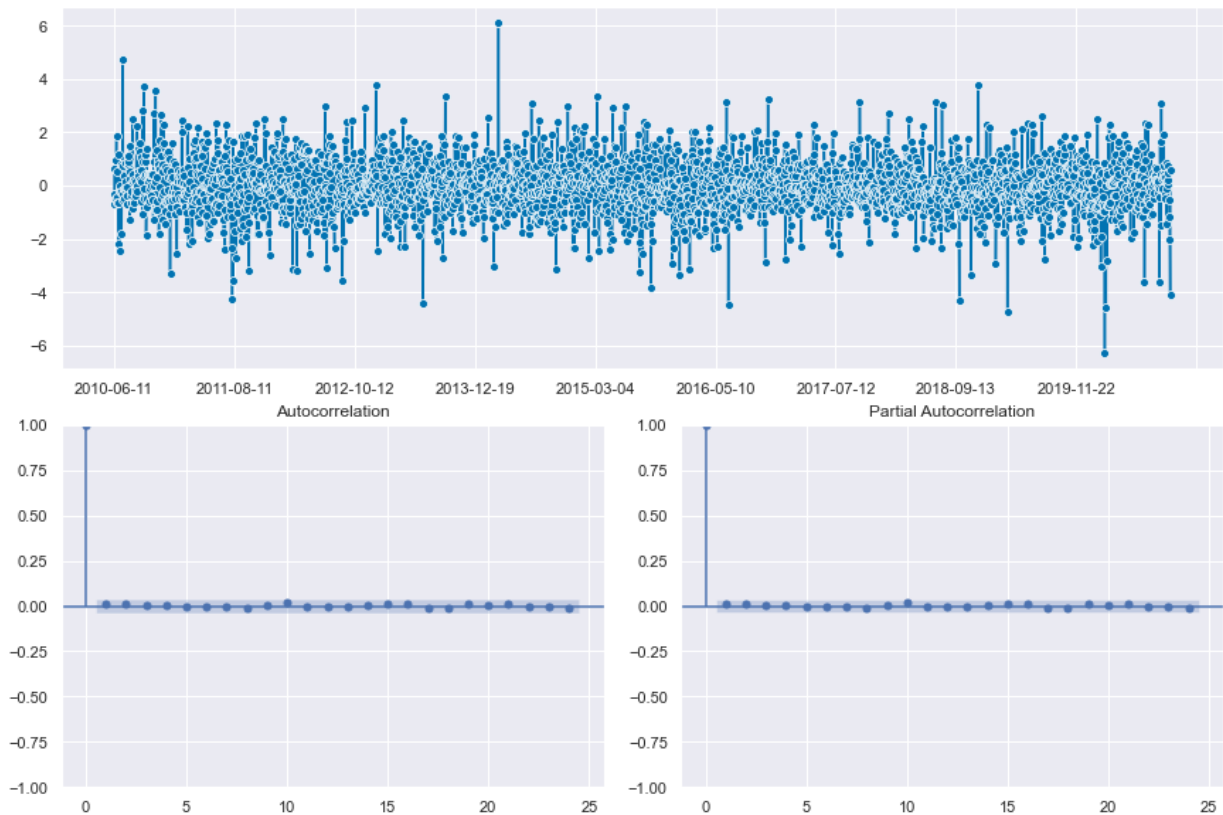


T STUDENT:

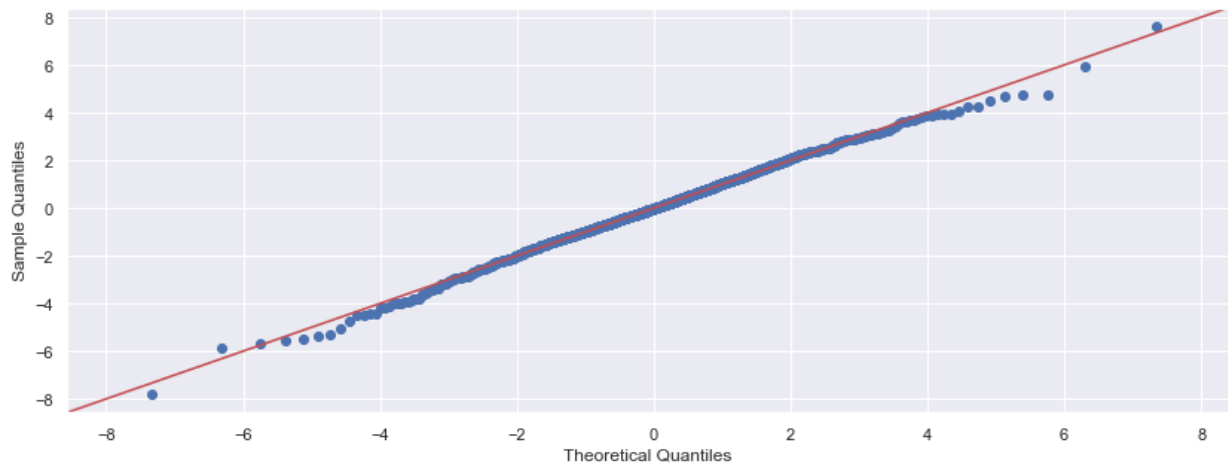
Igual que antes, vemos los residuos gráficamente:

Y comprobamos que siguen siendo ruido blanco

```
(<Figure size 864x576 with 3 Axes>,
 array([<AxesSubplot: >, <AxesSubplot: title='center': 'Autocorrelation'>,
        <AxesSubplot: title='center': 'Partial Autocorrelation'>],
 dtype=object))
```



Vemos que se ajusta mucho mejor a la recta.



6. Forecast noviembre 2020

Por último hemos hecho una predicción para noviembre, como nuestro último valor era octubre de 2020 incluido, solo hemos tenido que especificar un horizonte de 30 días. Nuestro res viene de la última ejecución de la T de Student, así que es el modelo que estamos usando para predecir.

	h.01	h.02	h.03	h.04	h.05	h.06	h.07	h.08	h.09	h.10	...	h.21	h.22	
Date														
2020-10-29	-1.013913	0.344524	-0.818159	-0.658345	-1.465898	-1.745747	1.09392	-2.194684	-0.801843	-0.434956	...	0.040997	-1.180094	-0.

1 rows × 30 columns