Predicting the oil price

Code

For reading the oil dataset we will start with loading the required Generative Additive Modeling (GAMLSS) module.

Hide

```
install.packages(c("gamlss","gamlss.add","gamlss.dist"))
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages('forecast', dependencies = TRUE)
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("ggplot2", dependencies=TRUE)
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages('GGally')
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("psych")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("rmarkdown")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("contrib.url")
```

```
WARNING: Rtools is required to build R packages but is not currently installed. Please downlo
ad and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/sry18/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
```

```
Warning in install.packages :
  package 'contrib.url' is not available for this version of R

A version of this package for your version of R might be available elsewhere,
see the ideas at
https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages

Restarting R session...


Restarting R session...
```

PREPARATION

Step 1: Read the oil dataset and keep the file locally available.

Hide

```
# Extract the oil dataset
data(oil)
View(oil)
```

Step 2: Data cleaning

Hide

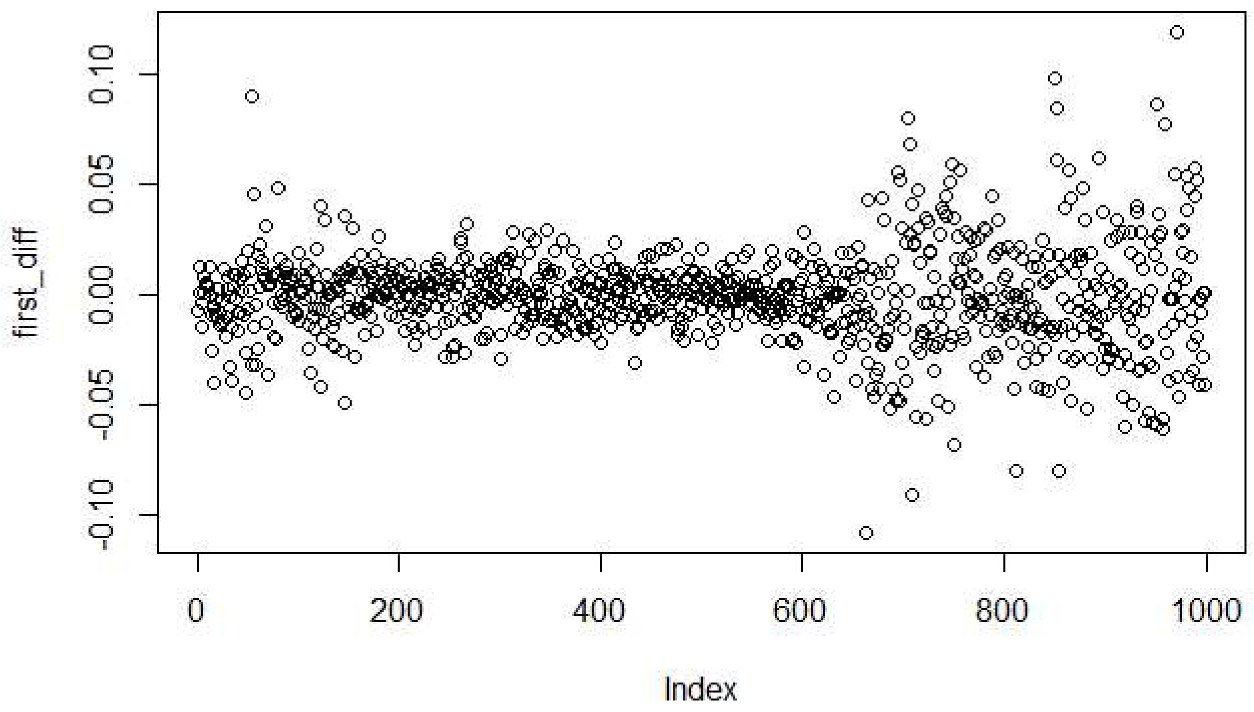
```
df <- na.omit(oil) # Method 1 - Remove NA
data_los <- as.character(nrow(oil) - nrow(df))
sprintf("Rows lost while removing null values are: %s", data_los)
```

```
[1] "Rows lost while removing null values are: 0"
```

1. EXPLORATORY DATA ANALYSIS Step 3: First only select the time lagged values and compute Partial Auto-Correlation Function to assert the impact of previous months oil prices on the current price. However as the oil price signal is non-stationary, we take the first-difference to make the time evolution stationary (the variance is however is not IID, can see a clear increase towards the end of the time-frame).

Hide

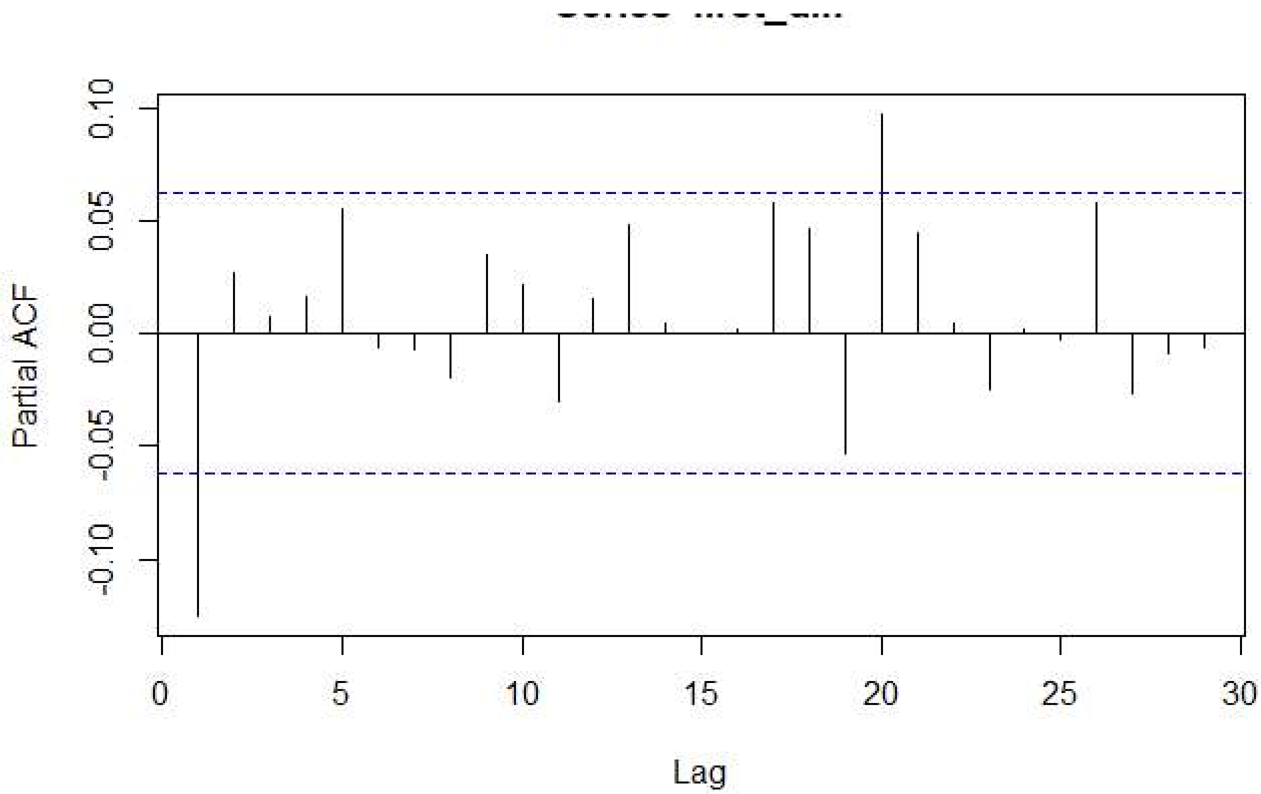
```
library(ggplot2)

cleaned_price_oil <- forecast::tsclean(df[,1])
first_diff <- diff(cleaned_price_oil)

# Plot the non-seasonal first-order difference of the oil price
plot(first_diff)
```



It can be observed that we have successfully removed the seasonality.

Hide

```
# Calculate the PACF to check for lag dependancy
pacf(first_diff)
```



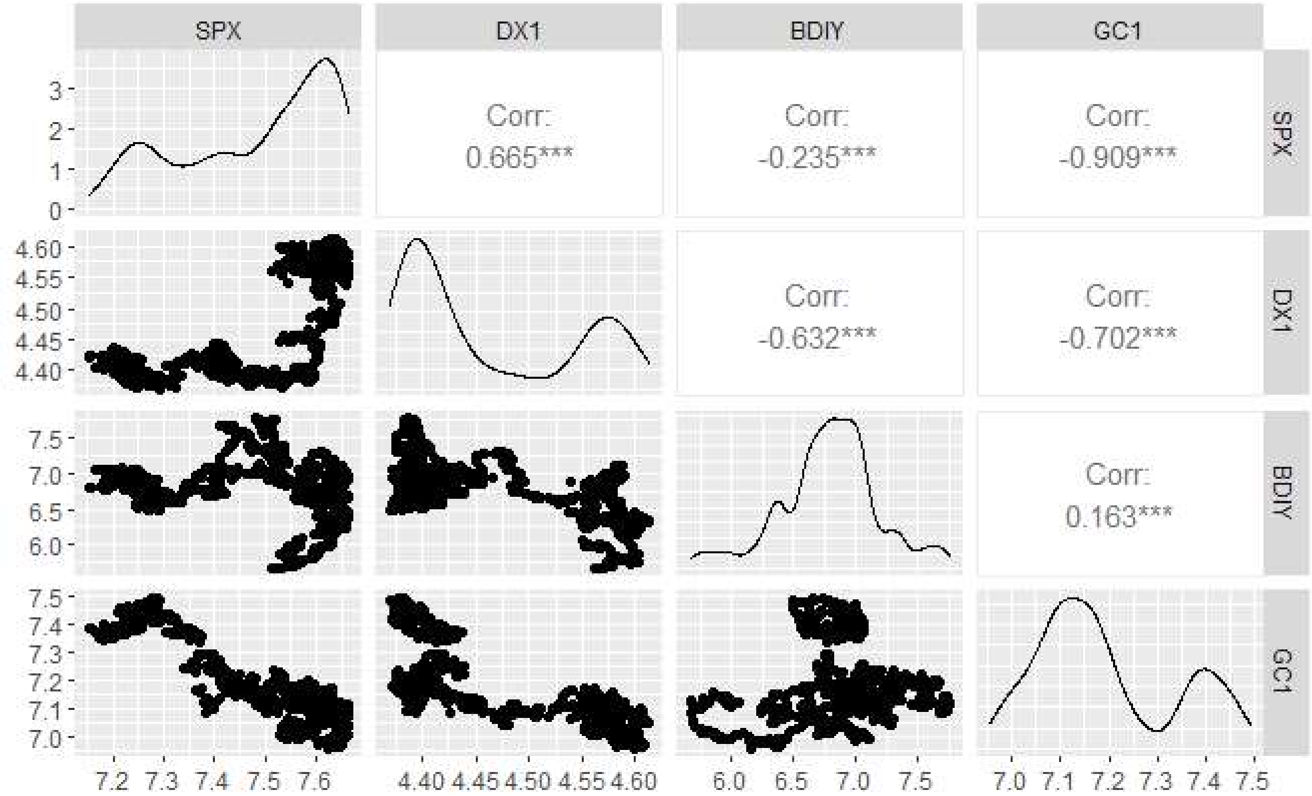
As noticed from the Partial Auto-Correlation function, except for the first lagged value, with 95% probability we can confirm that the influence of old oil price data is minimal on the current price.

We can also check the interdependence of other parameters and their distribution using the pair-wise plots.

Hide

```
#create pairs plot
# Let us only visualise the S & P Index, The US Dollar index, Baltic dry index, Gold price im
pact
df_variables = df[,16:25]

ggpairs(df_variables, columns = c("SPX_log", "DX1_log", "BDIY_log", "GC1_log"),
columnLabels = c("SPX", "DX1", "BDIY", "GC1"))
```

```
 plot: [1,1] [=====>-------------------------------------------------------------------------
------------]  6% est: 0s
 plot: [1,2] [==========>--------------------------------------------------------------------
------------] 12% est: 0s
 plot: [1,3] [================>--------------------------------------------------------------
------------] 19% est: 1s
 plot: [1,4] [======================>--------------------------------------------------------
------------] 25% est: 1s
 plot: [2,1] [===========================>---------------------------------------------------
------------] 31% est: 1s
 plot: [2,2] [=================================>---------------------------------------------
------------] 38% est: 0s
 plot: [2,3] [=======================================>---------------------------------------
------------] 44% est: 0s
 plot: [2,4] [=============================================>---------------------------------
------------] 50% est: 0s
 plot: [3,1] [==================================================>----------------------------
------------] 56% est: 0s
 plot: [3,2] [========================================================>----------------------
------------] 62% est: 0s
 plot: [3,3] [==============================================================>----------------
------------] 69% est: 0s
 plot: [3,4] [===================================================================>-----------
------------] 75% est: 0s
 plot: [4,1] [=========================================================================>-----
------------] 81% est: 0s
 plot: [4,2] [===============================================================================
>-----------] 88% est: 0s
 plot: [4,3] [===============================================================================
=====>------] 94% est: 0s
 plot: [4,4] [===============================================================================
============]100% est: 0s
```



Hide

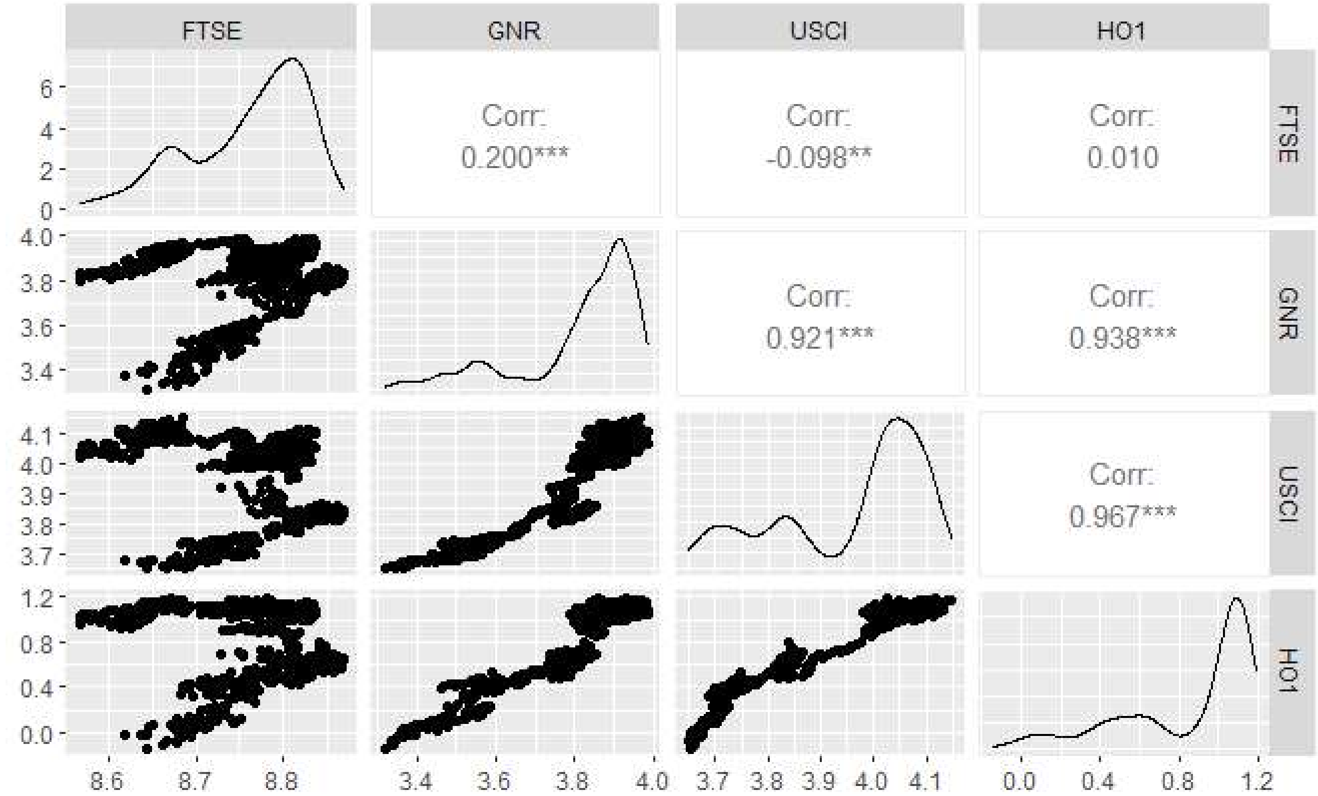
```
# ggpairs(df_other)
```

Gold price is observed to be negatively correlated with the S & P index - commodities investment could grow while markets tend bearish. However, as one could expect the S & P index and the US dollar are positively correlated. Similarly Baltic dry index and Dollar index are negatively correlated.

Hide

```
ggpairs(df_variables, columns = c("FTSE_log", "GNR_log", "USCI_log", "HO1_log"),
columnLabels = c("FTSE", "GNR", "USCI", "HO1"))
```

```
 plot: [1,1] [=====>-------------------------------------------------------------------------
------------]  6% est: 0s
 plot: [1,2] [==========>--------------------------------------------------------------------
------------] 12% est: 0s
 plot: [1,3] [================>--------------------------------------------------------------
------------] 19% est: 1s
 plot: [1,4] [======================>--------------------------------------------------------
------------] 25% est: 1s
 plot: [2,1] [===========================>---------------------------------------------------
------------] 31% est: 1s
 plot: [2,2] [=================================>---------------------------------------------
------------] 38% est: 0s
 plot: [2,3] [=======================================>---------------------------------------
------------] 44% est: 0s
 plot: [2,4] [=============================================>---------------------------------
------------] 50% est: 0s
 plot: [3,1] [==================================================>----------------------------
------------] 56% est: 0s
 plot: [3,2] [========================================================>----------------------
------------] 62% est: 0s
 plot: [3,3] [==============================================================>----------------
------------] 69% est: 0s
 plot: [3,4] [===================================================================>-----------
------------] 75% est: 0s
 plot: [4,1] [=========================================================================>-----
------------] 81% est: 0s
 plot: [4,2] [===============================================================================
>-----------] 88% est: 0s
 plot: [4,3] [===============================================================================
=====>------] 94% est: 0s
 plot: [4,4] [===============================================================================
============]100% est: 0s
```



A strong correlation exists amongst US Commodity exchange index, Global Natural Resource index and the heating oil value - as one can anticipate.

2. The distribution of the variables can be observed to be non-stationary however they are scaled logarthmically and hence do not require any further scaling/normalization.

It will be interesting however to check how the oil price correlates to these variables.

Hide

```
library(ggplot2)
par(mfrow = c(3, 3))

plot(df[,16], df[,1], xlab='Baltic Dry Index', ylab='Oil Price')
plot(df[,17], df[,1], xlab='S&P 500 index', ylab='Oil Price')
```

Hide

```
plot(df[,18], df[,1], xlab='US Dollar Index.', ylab='Oil Price')
plot(df[,19], df[,1], xlab='Traded gold price contract', ylab='Oil Price')
```

Hide

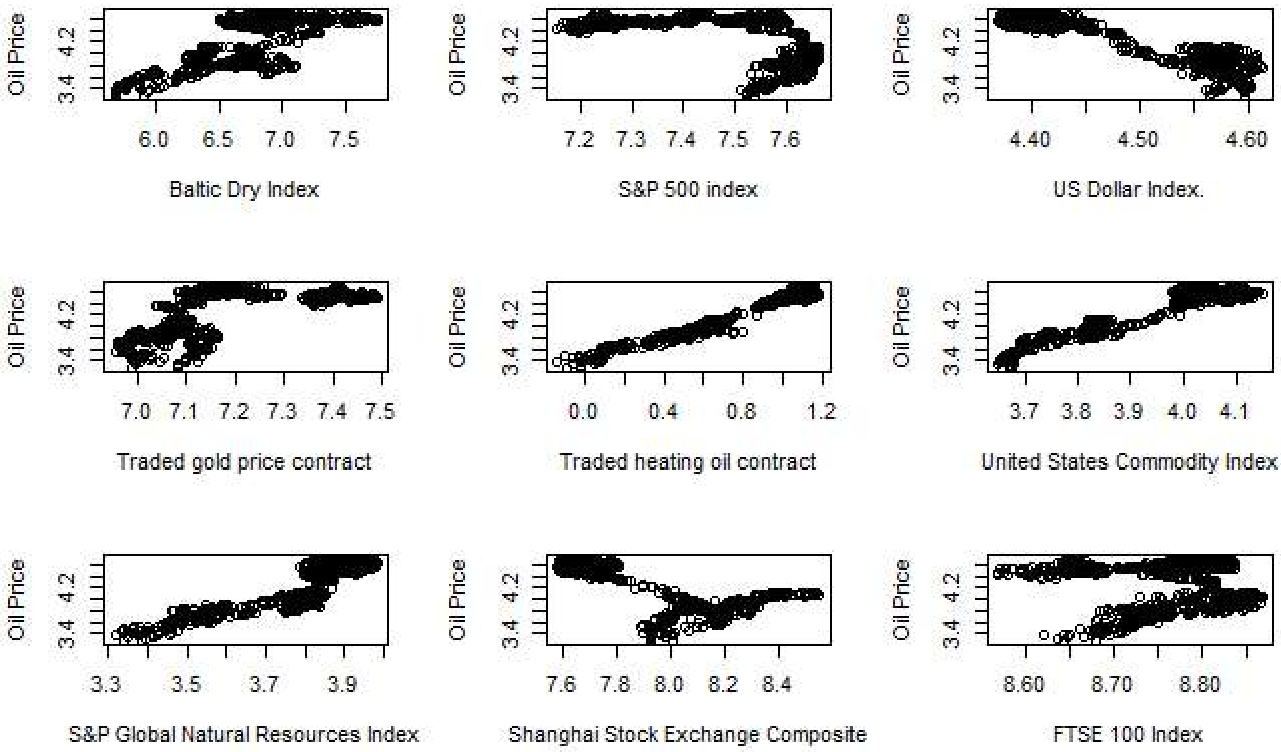
```


plot(df[,20], df[,1], xlab='Traded heating oil contract', ylab='Oil Price')
plot(df[,21], df[,1], xlab='United States Commodity Index', ylab='Oil Price')
```

Hide

```
plot(df[,22], df[,1], xlab='S&P Global Natural Resources Index', ylab='Oil Price')
plot(df[,23], df[,1], xlab='Shanghai Stock Exchange Composite', ylab='Oil Price')
```

Hide

```
plot(df[,24], df[,1], xlab='FTSE 100 Index', ylab='Oil Price')

x = df[,1]
par(mfrow = c(1, 1))
```



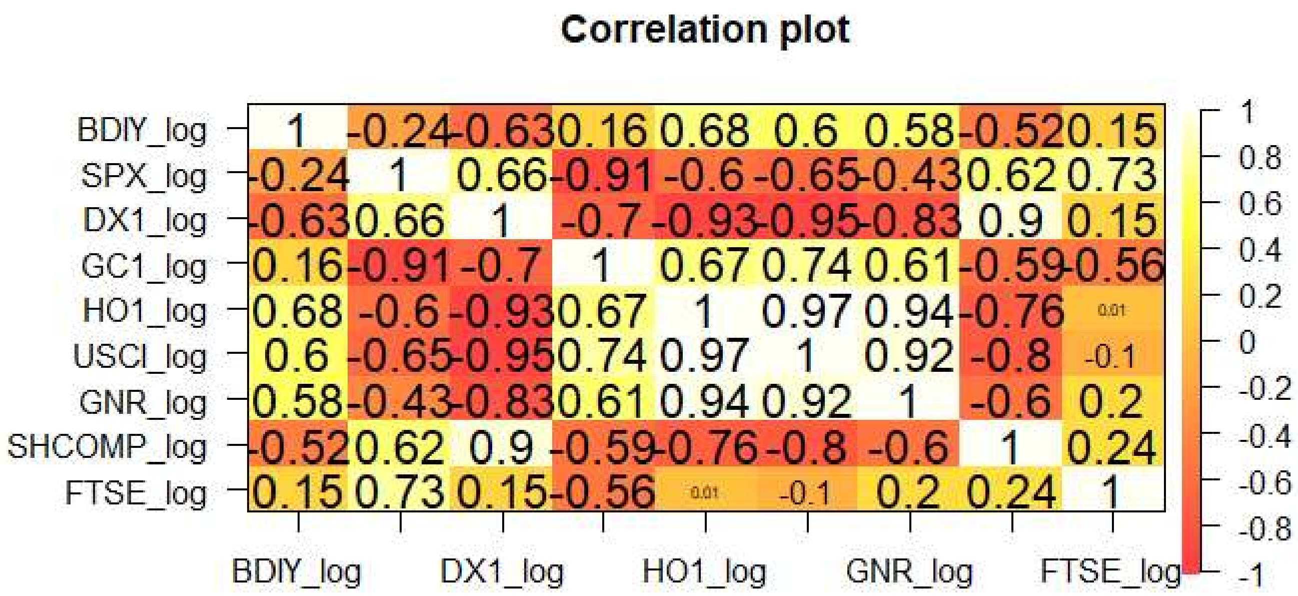
This plot heavily affects the degrees of freedom we choose downstream for the Generative Additive Models. It can be observed that the Oil price varies linearly with Traded heating oil contract, US Commodity index (positive) and US Dollar index (negative). With the gold price the evolution is more logarthmic in nature. It has a less clear non-linear behavior with BDI, S&P 500, FTSE, Shangai SE Composite.

The correlation amongst the variables can be better explained using the correlation heat map.

Correlation heat map

Hide

```
corPlot(df[, 16:24],
        gr = colorRampPalette(heat.colors(40)))
```



3. Initial Model development: We can consider the first lag value along with the above variables to develop the model and make the predictions.

Hide

```
# DF dataset cut off the past traded Oil contracts and keep only the lagged variable
idx_DF = sample(dim(df)[1] , 0.70*dim(df)[1] , replace = F)
training = df[idx_DF , ]
testing = df[-idx_DF , ]
X1 = df['OILPRICE']
```

Let’s see how the fuller model, consisting all the variables (this doesn’t necessarily translate to a better model).

Hide

```
# Model - Assuming linear dependance with USCI, DXI, HO1
df_o = oil[,-2:-24]

attach(df_o)
```

```
The following objects are masked from df_n:

    OILPRICE, respLAG

The following objects are masked from df_new (pos = 4):

    OILPRICE, respLAG

The following objects are masked from df_new (pos = 5):

    OILPRICE, respLAG

The following objects are masked from df_o (pos = 6):

    OILPRICE, respLAG
```

Hide

```
model <-
  gamlss(
    OILPRICE~ pbc(respLAG) + pbc(BDIY_log) + pbc(SPX_log) + DX1_log + pbc(GC1_log)
               + HO1_log + USCI_log + pbc(GNR_log) + pbc(SHCOMP_log) + pbc(FTSE_log) + pbc(CL
2_log)
    + pbc(CL3_log) + pbc(CL4_log) + pbc(CL5_log) + pbc(CL6_log) + pbc(CL7_log) + pbc(CL8_log)
+ pbc(CL9_log)
    + pbc(CL10_log) + pbc(CL11_log) + pbc(CL12_log) + pbc(CL13_log) + pbc(CL14_log) + pbc(CL1
5_log),
    family = BCPE,
    data = training, method = RS()
  )
```

```
GAMLSS-RS iteration 1: Global Deviance = -2246.04
GAMLSS-RS iteration 2: Global Deviance = -2951.502
GAMLSS-RS iteration 3: Global Deviance = -3147.04
GAMLSS-RS iteration 4: Global Deviance = -3204.629
GAMLSS-RS iteration 5: Global Deviance = -3218.39
GAMLSS-RS iteration 6: Global Deviance = -3219.642
GAMLSS-RS iteration 7: Global Deviance = -3219.029
GAMLSS-RS iteration 8: Global Deviance = -3215.459
GAMLSS-RS iteration 9: Global Deviance = -3217.4
GAMLSS-RS iteration 10: Global Deviance = -3221.864
GAMLSS-RS iteration 11: Global Deviance = -3208.508
GAMLSS-RS iteration 12: Global Deviance = -3203.195
GAMLSS-RS iteration 13: Global Deviance = -3193.928
GAMLSS-RS iteration 14: Global Deviance = -3180.435
GAMLSS-RS iteration 15: Global Deviance = -3180.214
GAMLSS-RS iteration 16: Global Deviance = -3154.747
GAMLSS-RS iteration 17: Global Deviance = -3156.837
GAMLSS-RS iteration 18: Global Deviance = -3143.092
GAMLSS-RS iteration 19: Global Deviance = -3144.463
GAMLSS-RS iteration 20: Global Deviance = -3152.757
```

```
Warning in RS() : Algorithm RS has not yet converged
```

Hide

```
)
```

```
Error: unexpected ')' in ")"
```

Now let’s consider a smaller set of variables based on our observation from the PACF plot, consisting only the most significant lags - only the 1st lag was found to be significant.

Hide

```


df_n = oil[,-2:-15]

attach(df_n)
```

```
The following objects are masked from df_new (pos = 3):

    BDIY_log, DX1_log, FTSE_log, GC1_log, GNR_log, HO1_log, OILPRICE, respLAG,
    SHCOMP_log, SPX_log, USCI_log

The following objects are masked from df_new (pos = 4):

    BDIY_log, DX1_log, FTSE_log, GC1_log, GNR_log, HO1_log, OILPRICE, respLAG,
    SHCOMP_log, SPX_log, USCI_log

The following objects are masked from df_o:

    OILPRICE, respLAG
```

Hide

```
model2 <-
  gamlss(
    OILPRICE ~ pbc(respLAG) + pbc(BDIY_log) + pbc(SPX_log) + DX1_log + pbc(GC1_log)
               + HO1_log + USCI_log + pbc(GNR_log) + pbc(SHCOMP_log) + pbc(FTSE_log),
    family = BCPE,
    data = training, method = RS()
  )
```

```
GAMLSS-RS iteration 1: Global Deviance = -2221.136
GAMLSS-RS iteration 2: Global Deviance = -3008.715
GAMLSS-RS iteration 3: Global Deviance = -3204.615
GAMLSS-RS iteration 4: Global Deviance = -3262.75
GAMLSS-RS iteration 5: Global Deviance = -3258.714
GAMLSS-RS iteration 6: Global Deviance = -3229.195
GAMLSS-RS iteration 7: Global Deviance = -3232.942
GAMLSS-RS iteration 8: Global Deviance = -3205.129
GAMLSS-RS iteration 9: Global Deviance = -3194.898
GAMLSS-RS iteration 10: Global Deviance = -3170.975
GAMLSS-RS iteration 11: Global Deviance = -3169.605
GAMLSS-RS iteration 12: Global Deviance = -3204.637
GAMLSS-RS iteration 13: Global Deviance = -3195.318
GAMLSS-RS iteration 14: Global Deviance = -3183.148
GAMLSS-RS iteration 15: Global Deviance = -3181.221
GAMLSS-RS iteration 16: Global Deviance = -3175.614
GAMLSS-RS iteration 17: Global Deviance = -3169.148
GAMLSS-RS iteration 18: Global Deviance = -3166.234
GAMLSS-RS iteration 19: Global Deviance = -3176.445
GAMLSS-RS iteration 20: Global Deviance = -3150.578
```

```
Warning in RS() : Algorithm RS has not yet converged
```

Hide

```
  )
```

```
Error: unexpected ')' in "  )"
```

4. It can be observed that the difference in accuracy for the 1st and the 2nd model is less and hence it is a better choice to consider less number of variables.

Hide

```
df_new = oil[,-2:-15]

attach(df_new)
```

```
The following objects are masked from df_o (pos = 3):

    OILPRICE, respLAG

The following objects are masked from df_n:

    BDIY_log, DX1_log, FTSE_log, GC1_log, GNR_log, HO1_log, OILPRICE, respLAG, SHCOMP_log, SP
X_log,
    USCI_log

The following objects are masked from df_new (pos = 5):

    BDIY_log, DX1_log, FTSE_log, GC1_log, GNR_log, HO1_log, OILPRICE, respLAG, SHCOMP_log, SP
X_log,
    USCI_log

The following objects are masked from df_new (pos = 6):

    BDIY_log, DX1_log, FTSE_log, GC1_log, GNR_log, HO1_log, OILPRICE, respLAG, SHCOMP_log, SP
X_log,
    USCI_log

The following objects are masked from df_o (pos = 7):

    OILPRICE, respLAG
```

Hide

```
model3 <-
  gamlss(
    OILPRICE ~ cs(respLAG, df=3) + cs(BDIY_log, df=3) + cs(SPX_log, df=3) + cs(DX1_log, df=3)
+ cs(GC1_log, df=3)
               + cs(HO1_log, df=3) + cs(USCI_log, df=3) + cs(GNR_log, df=3) + cs(SHCOMP_log,
 df=3) + cs(FTSE_log,  df=3),
    family = BCPE,
    data = training, method = RS()
  )
```

```
GAMLSS-RS iteration 1: Global Deviance = -2348.629
GAMLSS-RS iteration 2: Global Deviance = -3098.636
GAMLSS-RS iteration 3: Global Deviance = -3457.227
GAMLSS-RS iteration 4: Global Deviance = -3490.457
GAMLSS-RS iteration 5: Global Deviance = -3489.36
GAMLSS-RS iteration 6: Global Deviance = -3494.061
GAMLSS-RS iteration 7: Global Deviance = -3499.815
GAMLSS-RS iteration 8: Global Deviance = -3500.606
GAMLSS-RS iteration 9: Global Deviance = -3498.706
GAMLSS-RS iteration 10: Global Deviance = -3498.584
GAMLSS-RS iteration 11: Global Deviance = -3503.907
GAMLSS-RS iteration 12: Global Deviance = -3502.775
GAMLSS-RS iteration 13: Global Deviance = -3503.597
GAMLSS-RS iteration 14: Global Deviance = -3505.483
GAMLSS-RS iteration 15: Global Deviance = -3508.709
GAMLSS-RS iteration 16: Global Deviance = -3507.902
```

Now let us do some hyperparameter tuning to find the best degrees of freedom.

Hide

```
# 0 > DF PARAMETER RANGE =< 5.
val2 <- find.hyper(model3,parameters=c(1,5))
```

```
par 1 5 crit= -3414.63 with pen= 2
par 1.1 5 crit= -3414.63 with pen= 2
par 0.9 5 crit= -3414.63 with pen= 2
par 1 5.1 crit= -3414.63 with pen= 2
par 1 4.9 crit= -3414.63 with pen= 2
```

Hide

```
val2
```

```
$par
[1] 1 5

$value
[1] -3414.63

$counts
function gradient
       1        1

$convergence
[1] 0

$message
[1] "CONVERGENCE: NORM OF PROJECTED GRADIENT <= PGTOL"
```

Refined model3 with the updated degrees of freedom which as per hyperparameter tuning suggests 5.

Hide

```
df_new = oil[,-2:-15]

attach(df_new)
```

```
The following objects are masked from df_new (pos = 3):

    BDIY_log, DX1_log, FTSE_log, GC1_log, GNR_log, HO1_log, OILPRICE, respLAG, SHCOMP_log, SP
X_log,
    USCI_log

The following objects are masked from df_o (pos = 4):

    OILPRICE, respLAG

The following objects are masked from df_n:

    BDIY_log, DX1_log, FTSE_log, GC1_log, GNR_log, HO1_log, OILPRICE, respLAG, SHCOMP_log, SP
X_log,
    USCI_log

The following objects are masked from df_new (pos = 6):

    BDIY_log, DX1_log, FTSE_log, GC1_log, GNR_log, HO1_log, OILPRICE, respLAG, SHCOMP_log, SP
X_log,
    USCI_log

The following objects are masked from df_new (pos = 7):

    BDIY_log, DX1_log, FTSE_log, GC1_log, GNR_log, HO1_log, OILPRICE, respLAG, SHCOMP_log, SP
X_log,
    USCI_log

The following objects are masked from df_o (pos = 8):

    OILPRICE, respLAG
```

Hide

```
model4 <-
  gamlss(
    OILPRICE ~ cs(respLAG, df=5) + cs(BDIY_log, df=5) + cs(SPX_log, df=5) + cs(DX1_log, df=5)
+ cs(GC1_log, df=5)
               + cs(HO1_log, df=5) + cs(USCI_log, df=5) + cs(GNR_log, df=5) + cs(SHCOMP_log,
 df=5) + cs(FTSE_log,  df=5),
    family = BCPE,
    data = training, method = RS()
  )
```

```
GAMLSS-RS iteration 1: Global Deviance = -2381.827
GAMLSS-RS iteration 2: Global Deviance = -3090.493
GAMLSS-RS iteration 3: Global Deviance = -3450.339
GAMLSS-RS iteration 4: Global Deviance = -3537.017
GAMLSS-RS iteration 5: Global Deviance = -3526.956
GAMLSS-RS iteration 6: Global Deviance = -3524.793
GAMLSS-RS iteration 7: Global Deviance = -3521.091
GAMLSS-RS iteration 8: Global Deviance = -3520.61
GAMLSS-RS iteration 9: Global Deviance = -3520.128
GAMLSS-RS iteration 10: Global Deviance = -3518.087
GAMLSS-RS iteration 11: Global Deviance = -3520.801
GAMLSS-RS iteration 12: Global Deviance = -3521.095
GAMLSS-RS iteration 13: Global Deviance = -3521.164
GAMLSS-RS iteration 14: Global Deviance = -3519.484
GAMLSS-RS iteration 15: Global Deviance = -3522.963
GAMLSS-RS iteration 16: Global Deviance = -3523.081
GAMLSS-RS iteration 17: Global Deviance = -3528.866
GAMLSS-RS iteration 18: Global Deviance = -3526.809
GAMLSS-RS iteration 19: Global Deviance = -3525.422
GAMLSS-RS iteration 20: Global Deviance = -3526.272
```

```
Warning in RS() : Algorithm RS has not yet converged
```

It can now be observed that based on the inputs from the hyper-parameter tuning the non-penalised cubic spline method better modelled the oil price multivariate problem with a Global Deviance (-2*max_log_likelihood of -3536 units). Beyond 5 could make the model to overfit to the exisiting data.

Hide

```
summary(model)
```

```
******************************************************************
Family:  c("BCPE", "Box-Cox Power Exponential") 

Call:  gamlss(formula = OILPRICE ~ pbc(respLAG) + pbc(BDIY_log) +  
    pbc(SPX_log) + DX1_log + pbc(GC1_log) + HO1_log +      USCI_log + pbc(GNR_log) + pbc(SHCO
MP_log) + pbc(FTSE_log) +  
    pbc(CL2_log) + pbc(CL3_log) + pbc(CL4_log) + pbc(CL5_log) +  
    pbc(CL6_log) + pbc(CL7_log) + pbc(CL8_log) + pbc(CL9_log) +      pbc(CL10_log) + pbc(CL11
_log) + pbc(CL12_log) +  
    pbc(CL13_log) + pbc(CL14_log) + pbc(CL15_log),      family = BCPE, data = training, metho
d = RS()) 

Fitting method: RS() 

------------------------------------------------------------------
Mu link function:  identity
Mu Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.8276151  0.0003633   10537   <2e-16 ***
DX1_log     -0.3819684  0.0002605   -1466   <2e-16 ***
HO1_log      0.5645356  0.0004038    1398   <2e-16 ***
USCI_log     0.4247589  0.0002737    1552   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
Sigma link function:  log
Sigma Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -4.3580     0.1503  -28.99   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
Nu link function:  identity 
Nu Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)    74.57      11.12   6.705  6.4e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
Tau link function:  log
Tau Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.84159    0.08682  -9.694   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
NOTE: Additive smoothing terms exist in the formulas: 
 i) Std. Error for smoothers are for the linear effect only. 
ii) Std. Error for the linear terms maybe are not accurate. 
------------------------------------------------------------------
No. of observations in the fit:  700 
```

```
Degrees of Freedom for the fit:  274.3295
      Residual Deg. of Freedom:  425.6705
                      at cycle:  20

Global Deviance:     -3152.757
            AIC:     -2604.098
            SBC:     -1355.602
******************************************************************
```

Hide

```
summary(model2)
```

```
******************************************************************
Family:  c("BCPE", "Box-Cox Power Exponential") 

Call:  gamlss(formula = OILPRICE ~ pbc(respLAG) + pbc(BDIY_log) +  
    pbc(SPX_log) + DX1_log + pbc(GC1_log) + HO1_log +      USCI_log + pbc(GNR_log) + pbc(SHCO
MP_log) + pbc(FTSE_log),  
    family = BCPE, data = training, method = RS()) 

Fitting method: RS() 

------------------------------------------------------------------
Mu link function:  identity
Mu Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.4857810  0.0002828   12326   <2e-16 ***
DX1_log     -0.3546869  0.0002030   -1748   <2e-16 ***
HO1_log      0.5374866  0.0002954    1819   <2e-16 ***
USCI_log     0.4849047  0.0002153    2253   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
Sigma link function:  log
Sigma Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -4.0370     0.1769  -22.82   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
Nu link function:  identity 
Nu Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   67.378      9.964   6.762 3.35e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
Tau link function:  log
Tau Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.06737    0.08259  -12.92   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
NOTE: Additive smoothing terms exist in the formulas: 
 i) Std. Error for smoothers are for the linear effect only. 
ii) Std. Error for the linear terms maybe are not accurate. 
------------------------------------------------------------------
No. of observations in the fit:  700 
Degrees of Freedom for the fit:  125.981
      Residual Deg. of Freedom:  574.019 
                      at cycle:  20 
```

```
Global Deviance:     -3150.578
            AIC:     -2898.616
            SBC:     -2325.266
******************************************************************
```

Hide

```
summary(model3)
```

```
******************************************************************
Family:  c("BCPE", "Box-Cox Power Exponential")

Call:  gamlss(formula = OILPRICE ~ cs(respLAG, df = 3) + cs(BDIY_log,
    df = 3) + cs(SPX_log, df = 3) + cs(DX1_log, df = 3) +      cs(GC1_log, df = 3) + cs(HO1_l
og, df = 3) + cs(USCI_log,
    df = 3) + cs(GNR_log, df = 3) + cs(SHCOMP_log,      df = 3) + cs(FTSE_log, df = 3), famil
y = BCPE,
    data = training, method = RS())

Fitting method: RS()

------------------------------------------------------------------
Mu link function:  identity
Mu Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)             0.224348   0.001486 151.020  < 2e-16 ***
cs(respLAG, df = 3)     0.980853   0.001875 523.078  < 2e-16 ***
cs(BDIY_log, df = 3)   -0.002937   0.001525  -1.926  0.05454 .
cs(SPX_log, df = 3)    -0.106373   0.004533 -23.465  < 2e-16 ***
cs(DX1_log, df = 3)     0.087663   0.002021  43.381  < 2e-16 ***
cs(GC1_log, df = 3)    -0.047789   0.003440 -13.892  < 2e-16 ***
cs(HO1_log, df = 3)    -0.009346   0.002118  -4.413 1.19e-05 ***
cs(USCI_log, df = 3)    0.016937   0.004544   3.727  0.00021 ***
cs(GNR_log, df = 3)     0.035970   0.004159   8.649  < 2e-16 ***
cs(SHCOMP_log, df = 3) -0.031727   0.003639  -8.718  < 2e-16 ***
cs(FTSE_log, df = 3)    0.077371   0.001327  58.317  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
Sigma link function:  log
Sigma Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.14648    0.05973  -86.16   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
Nu link function:  identity
Nu Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -7.415      8.536  -0.869    0.385

------------------------------------------------------------------
Tau link function:  log
Tau Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.40865    0.06847  -5.968 3.93e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
NOTE: Additive smoothing terms exist in the formulas:
 i) Std. Error for smoothers are for the linear effect only.
```

```
ii) Std. Error for the linear terms maybe are not accurate.
------------------------------------------------------------------
No. of observations in the fit:  700
Degrees of Freedom for the fit:  44.00114
      Residual Deg. of Freedom:  655.9989
                      at cycle:  20

Global Deviance:     -3502.633
            AIC:     -3414.63
            SBC:     -3214.378
******************************************************************
```

Hide

```
summary(model4)
```

```
Warning in summary.gamlss(model4) :
  summary: vcov has failed, option qr is used instead
```

```
******************************************************************
Family:  c("BCPE", "Box-Cox Power Exponential") 

Call:  gamlss(formula = OILPRICE ~ cs(respLAG, df = 5) + cs(BDIY_log,  
    df = 5) + cs(SPX_log, df = 5) + cs(DX1_log, df = 5) + cs(GC1_log,  
    df = 5) + cs(HO1_log, df = 5) + cs(USCI_log, df = 5) + cs(GNR_log,  
    df = 5) + cs(SHCOMP_log, df = 5) + cs(FTSE_log, df = 5),      family = BCPE, data = train
ing, method = RS()) 

Fitting method: RS() 

------------------------------------------------------------------
Mu link function:  identity
Mu Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)            -0.4766286  0.0728685  -6.541 1.25e-10 ***
cs(respLAG, df = 5)     0.9753722  0.0030854 316.126  < 2e-16 ***
cs(BDIY_log, df = 5)    0.0026403  0.0005703   4.630 4.44e-06 ***
cs(SPX_log, df = 5)    -0.0715082  0.0040504 -17.654  < 2e-16 ***
cs(DX1_log, df = 5)     0.1393390  0.0090132  15.459  < 2e-16 ***
cs(GC1_log, df = 5)    -0.0222988  0.0038068  -5.858 7.52e-09 ***
cs(HO1_log, df = 5)    -0.0189198  0.0038467  -4.918 1.11e-06 ***
cs(USCI_log, df = 5)    0.0445758  0.0065729   6.782 2.71e-11 ***
cs(GNR_log, df = 5)     0.0482589  0.0066275   7.282 9.73e-13 ***
cs(SHCOMP_log, df = 5) -0.0418713  0.0020828 -20.103  < 2e-16 ***
cs(FTSE_log, df = 5)    0.0707974  0.0076352   9.273  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
Sigma link function:  log
Sigma Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -5.11791    0.04825  -106.1   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
Nu link function:  identity 
Nu Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -4.697      7.490  -0.627    0.531

------------------------------------------------------------------
Tau link function:  log
Tau Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.49527    0.04925  -10.06   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

------------------------------------------------------------------
NOTE: Additive smoothing terms exist in the formulas: 
 i) Std. Error for smoothers are for the linear effect only. 
ii) Std. Error for the linear terms may not be reliable. 
```

```
------------------------------------------------------------------
No. of observations in the fit:  700
Degrees of Freedom for the fit:  63.99718
      Residual Deg. of Freedom:  636.0028
                      at cycle:  20

Global Deviance:     -3526.272
            AIC:     -3398.278
            SBC:     -3107.021
******************************************************************
```

Hide

```
testing$prediction3 <- predict(model3, newdata=testing[,-26], type = "response")
```

```
Warning in predict.gamlss(model3, newdata = testing[, -26], type = "response") :
  There is a discrepancy  between the original and the re-fit
 used to achieve 'safe' predictions
```

```
new prediction
```

Hide

```
testing$prediction4 <- predict(model4, newdata=testing[,-26], type = "response")
```

```
Warning in predict.gamlss(model4, newdata = testing[, -26], type = "response") :
  There is a discrepancy  between the original and the re-fit
 used to achieve 'safe' predictions
```

```
new prediction
```

Hide

```
# Let us compare log of the errors for the models3 and 4 when applied to the testing data
error3 <- log( sum( (testing[12] - testing[11])^2/(testing[11]^2) ), base=10 )
error4 <- log( sum( (testing[13] - testing[11])^2/(testing[11]^2) ), base=10 )
error3
```

```
[1] -3.223521
```

Hide

```
error4
```

```
[1] -2.646041
```

5. It can be observed that model 3 has smaller error when compared to original price.

To improve the model I would utilise Bayesian inference pipelines to identify what are the dominant confounding variables affecting the oil price. This will separate correlated effects from the causal and can hence provide a deeper understanding of the 'effects' on the oil price.

Further, Directed Acyclic Graphs can be generated to better visualise such confounding effects.

CORRELATION DOESN'T IMPLY CAUSATION…….

Hide

```
library(rmarkdown)
render("1-example.Rmd")
```