Coding test_Argus

```
library(gamlss)
```

```
## Loading required package: splines
```

```
## Loading required package: gamlss.data
```

```
##
## Attaching package: 'gamlss.data'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
## Loading required package: gamlss.dist
```

```
## Loading required package: MASS
```

```
## Loading required package: nlme
```

```
## Loading required package: parallel
```

```
##  **********   GAMLSS Version 5.3-4  **********
```

```
## For more on GAMLSS look at https://www.gamlss.com/
```

```
## Type gamlssNews() to see new features/changes/bug fixes.
```

```
library(gamlss.add)
```

```
## Loading required package: mgcv
```

```
## This is mgcv 1.8-35. For overview type 'help("mgcv-package")'.
```

```
## Loading required package: nnet
```

```
##
## Attaching package: 'nnet'
```

```
## The following object is masked from 'package:mgcv':
##
##     multinom
```

```
## Loading required package: rpart
```

```
library(gamlss.dist)
library(DT)
library(roll)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:nlme':
##
##     collapse
```

```
## The following object is masked from 'package:MASS':
##
##     select
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stats)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(ggpubr)
```

```
## Loading required package: ggplot2
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
## The following object is masked from 'package:gamlss.add':
##
##     tr
## The following object is masked from 'package:gamlss':
##
##     cs
```

```
library(magrittr)
```

```
# import data

oil_data <- gamlss.data::oil
summary(oil_data)
```

```
##     OILPRICE        CL2_log         CL3_log         CL4_log
##  Min.   :3.266   Min.   :3.345   Min.   :3.391   Min.   :3.428
##  1st Qu.:3.966   1st Qu.:3.982   1st Qu.:4.001   1st Qu.:4.017
##  Median :4.517   Median :4.519   Median :4.519   Median :4.518
##  Mean   :4.309   Mean   :4.317   Mean   :4.322   Mean   :4.326
##  3rd Qu.:4.580   3rd Qu.:4.581   3rd Qu.:4.581   3rd Qu.:4.579
##  Max.   :4.705   Max.   :4.696   Max.   :4.682   Max.   :4.672
##     CL5_log         CL6_log         CL7_log         CL8_log
##  Min.   :3.482   Min.   :3.501   Min.   :3.516   Min.   :3.529
##  1st Qu.:4.032   1st Qu.:4.048   1st Qu.:4.058   1st Qu.:4.068
##  Median :4.519   Median :4.518   Median :4.517   Median :4.514
##  Mean   :4.329   Mean   :4.331   Mean   :4.332   Mean   :4.333
##  3rd Qu.:4.575   3rd Qu.:4.570   3rd Qu.:4.563   3rd Qu.:4.556
```

```
##  Max.   :4.672   Max.   :4.673   Max.   :4.673   Max.   :4.673
##     CL9_log         CL10_log        CL11_log        CL12_log
##  Min.   :3.542   Min.   :3.555   Min.   :3.566   Min.   :3.576
##  1st Qu.:4.075   1st Qu.:4.082   1st Qu.:4.090   1st Qu.:4.097
##  Median :4.512   Median :4.509   Median :4.506   Median :4.503
##  Mean   :4.333   Mean   :4.337   Mean   :4.333   Mean   :4.332
##  3rd Qu.:4.550   3rd Qu.:4.546   3rd Qu.:4.542   3rd Qu.:4.537
##  Max.   :4.672   Max.   :4.670   Max.   :4.667   Max.   :4.663
##     CL13_log        CL14_log        CL15_log        BDIY_log
##  Min.   :3.585   Min.   :3.594   Min.   :3.603   Min.   :5.670
##  1st Qu.:4.103   1st Qu.:4.110   1st Qu.:4.117   1st Qu.:6.596
##  Median :4.500   Median :4.497   Median :4.493   Median :6.806
##  Mean   :4.332   Mean   :4.331   Mean   :4.331   Mean   :6.787
##  3rd Qu.:4.532   3rd Qu.:4.527   3rd Qu.:4.522   3rd Qu.:7.011
##  Max.   :4.658   Max.   :4.654   Max.   :4.649   Max.   :7.757
##     SPX_log         DX1_log         GC1_log         HO1_log
##  Min.   :7.153   Min.   :4.369   Min.   :6.956   Min.   :-0.1442
##  1st Qu.:7.354   1st Qu.:4.391   1st Qu.:7.089   1st Qu.: 0.6220
##  Median :7.531   Median :4.417   Median :7.159   Median : 1.0547
##  Mean   :7.481   Mean   :4.459   Mean   :7.192   Mean   : 0.8600
##  3rd Qu.:7.611   3rd Qu.:4.557   3rd Qu.:7.345   3rd Qu.: 1.1013
##  Max.   :7.664   Max.   :4.613   Max.   :7.491   Max.   : 1.1877
##     USCI_log        GNR_log        SHCOMP_log       FTSE_log
##  Min.   :3.650   Min.   :3.317   Min.   :7.576   Min.   :8.568
##  1st Qu.:3.838   1st Qu.:3.787   1st Qu.:7.652   1st Qu.:8.716
##  Median :4.021   Median :3.868   Median :7.734   Median :8.778
##  Mean   :3.962   Mean   :3.818   Mean   :7.840   Mean   :8.760
##  3rd Qu.:4.070   3rd Qu.:3.920   3rd Qu.:8.032   3rd Qu.:8.813
##  Max.   :4.148   Max.   :3.985   Max.   :8.550   Max.   :8.868
##     respLAG
##  Min.   :3.266
##  1st Qu.:3.966
##  Median :4.517
##  Mean   :4.310
##  3rd Qu.:4.580
##  Max.   :4.705
```

```
paste0("Oil dataset has ", dim(oil_data)[1], " observations and  ", dim(oil_data)[2], " variables. ")
```

```
## [1] "Oil dataset has 1000 observations and  25 variables. "
```

```
df <- as.data.frame(oil_data$OILPRICE)
oil_data1 <- oil_data$OILPRICE
oil_data2 <- oil_data$OILPRICE
oil_data2 <- as.matrix(oil_data2)
# rtolling standard deviation with window = 5
roll_sd_dev <- roll::roll_sd(oil_data2, 5)
df$roll_sd_dev <- roll_sd_dev
# Rolling mean with window = 5
roll_mean <- roll::roll_mean(oil_data2, 5)
df$roll_mean <- roll_mean
# Lagging with order = 2
lag1 <- dplyr::lag(oil_data1,2)
df$lag1<-lag1
# Leading with order = 2
```

```
lead <- dplyr::lead(oil_data1,2)
df$lead<-lead
Diff <- oil_data1 %>% diff()
Diff[1000] <- NA
df$diff <- Diff

paste0("df has ", dim(df)[1], " observations and  ", dim(df)[2], " variables. ")
```

```
## [1] "df has 1000 observations and  6 variables. "
```

```
head(df,n=10)
```

```
##    oil_data$OILPRICE roll_sd_dev roll_mean     lag1     lead          diff
## 1           4.640923          NA        NA       NA 4.634049 -0.0078462165
## 2           4.633077          NA        NA       NA 4.646312  0.0009720063
## 3           4.634049          NA        NA 4.640923 4.631520  0.0122629838
## 4           4.646312          NA        NA 4.633077 4.627616 -0.0147921680
## 5           4.631520 0.006246603  4.637176 4.634049 4.635214 -0.0039035865
## 6           4.627616 0.007035944  4.634515 4.646312 4.635796  0.0075979325
## 7           4.635214 0.006991536  4.634942 4.631520 4.640055  0.0005820722
## 8           4.635796 0.006979384  4.635292 4.627616 4.645544  0.0042582083
## 9           4.640055 0.004697145  4.634040 4.635214 4.649665  0.0054894923
## 10          4.645544 0.006612521  4.636845 4.635796 4.653293  0.0041213457
```

```
dfo<-df[, c("roll_sd_dev", "roll_mean")]
df1 <- as.data.frame(dfo)
Ratio <- df1[,1]/df1[,2]
df1$Ratio<-Ratio
Product <- df1[,1] * df1[,2]
df1$Product<- Product
head(df1, n = 10)
```

```
##    roll_sd_dev roll_mean       Ratio    Product
## 1           NA        NA          NA         NA
## 2           NA        NA          NA         NA
## 3           NA        NA          NA         NA
## 4           NA        NA          NA         NA
## 5  0.006246603  4.637176 0.001347070 0.02896660
## 6  0.007035944  4.634515 0.001518162 0.03260819
## 7  0.006991536  4.634942 0.001508441 0.03240537
## 8  0.006979384  4.635292 0.001505705 0.03235148
## 9  0.004697145  4.634040 0.001013618 0.02176676
## 10 0.006612521  4.636845 0.001426082 0.03066124
```

```
dfo2<-oil_data[,c("OILPRICE", "respLAG")]
df2 <- as.data.frame(dfo2)
diff <- (df2$OILPRICE - df2$respLAG)
df2$difference <- diff
diff <- as.matrix(diff)
roll_std <- roll::roll_sd(diff, 5)
comp_trans <- roll_std
df2$comp_trans<-comp_trans
head(df2, n = 10)
```

```
##    OILPRICE  respLAG    difference  comp_trans
## 1  4.640923 4.631812  0.0091112388          NA
```

```
## 2  4.633077 4.640923 -0.0078462165          NA
## 3  4.634049 4.633077  0.0009720063          NA
## 4  4.646312 4.634049  0.0122629838          NA
## 5  4.631520 4.646312 -0.0147921680 0.011343440
## 6  4.627616 4.631520 -0.0039035865 0.010142975
## 7  4.635214 4.627616  0.0075979325 0.010514120
## 8  4.635796 4.635214  0.0005820722 0.010510516
## 9  4.640055 4.635796  0.0042582083 0.008695036
## 10 4.645544 4.640055  0.0054894923 0.004534232
```

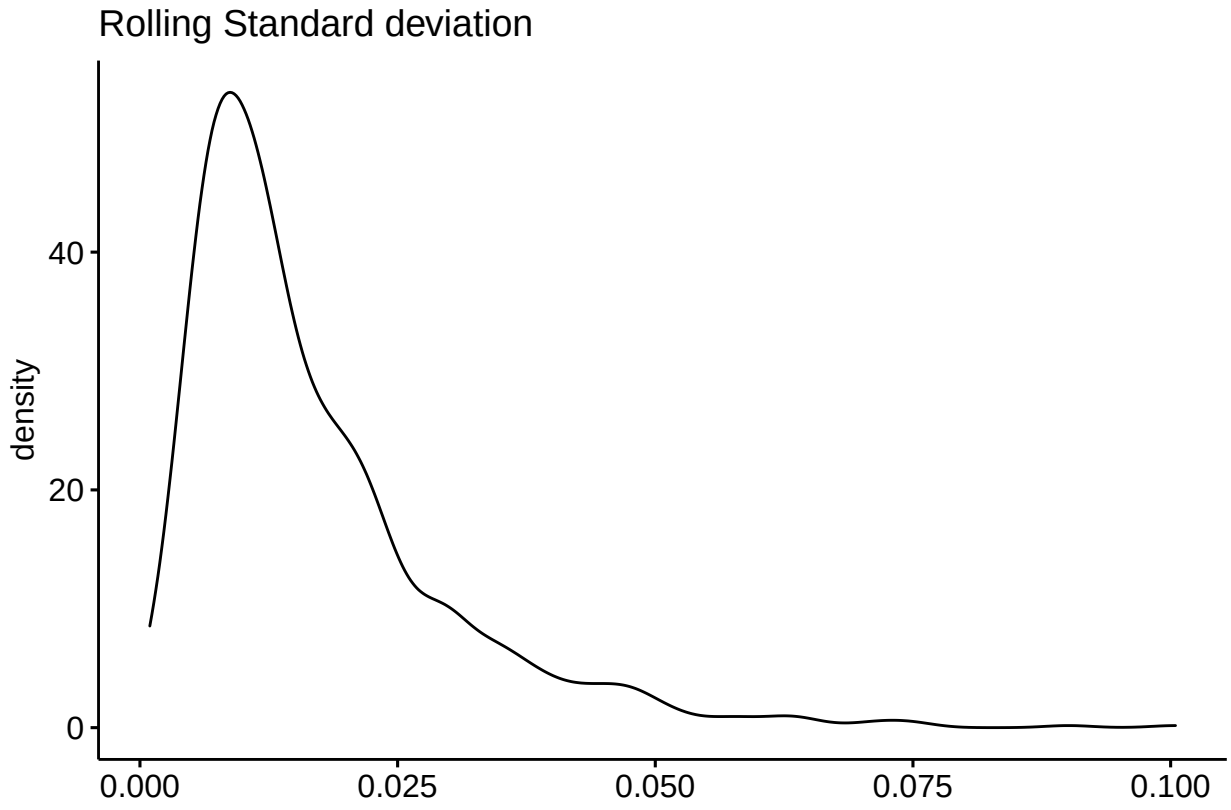
```
drivers <- cbind(df, df1, df2)
drivers <- drivers[, c("oil_data$OILPRICE", "roll_sd_dev", "roll_mean", "lag1", "lead", "diff", "Ratio"

head(drivers, n = 10)
```

```
##    oil_data$OILPRICE roll_sd_dev roll_mean     lag1     lead          diff
## 1           4.640923          NA        NA       NA 4.634049 -0.0078462165
## 2           4.633077          NA        NA       NA 4.646312  0.0009720063
## 3           4.634049          NA        NA 4.640923 4.631520  0.0122629838
## 4           4.646312          NA        NA 4.633077 4.627616 -0.0147921680
## 5           4.631520 0.006246603  4.637176 4.634049 4.635214 -0.0039035865
## 6           4.627616 0.007035944  4.634515 4.646312 4.635796  0.0075979325
## 7           4.635214 0.006991536  4.634942 4.631520 4.640055  0.0005820722
## 8           4.635796 0.006979384  4.635292 4.627616 4.645544  0.0042582083
## 9           4.640055 0.004697145  4.634040 4.635214 4.649665  0.0054894923
## 10          4.645544 0.006612521  4.636845 4.635796 4.653293  0.0041213457
##          Ratio    Product  comp_trans
## 1           NA         NA          NA
## 2           NA         NA          NA
## 3           NA         NA          NA
## 4           NA         NA          NA
## 5  0.001347070 0.02896660 0.011343440
## 6  0.001518162 0.03260819 0.010142975
## 7  0.001508441 0.03240537 0.010514120
## 8  0.001505705 0.03235148 0.010510516
## 9  0.001013618 0.02176676 0.008695036
## 10 0.001426082 0.03066124 0.004534232
```

```
# check normality
input_sd_dev<-drivers$roll_sd_dev
 print(ggdensity(input_sd_dev,
          main = "Rolling Standard deviation",
          xlab = ""))
```

```
## Warning: Removed 4 rows containing non-finite values (stat_density).
```



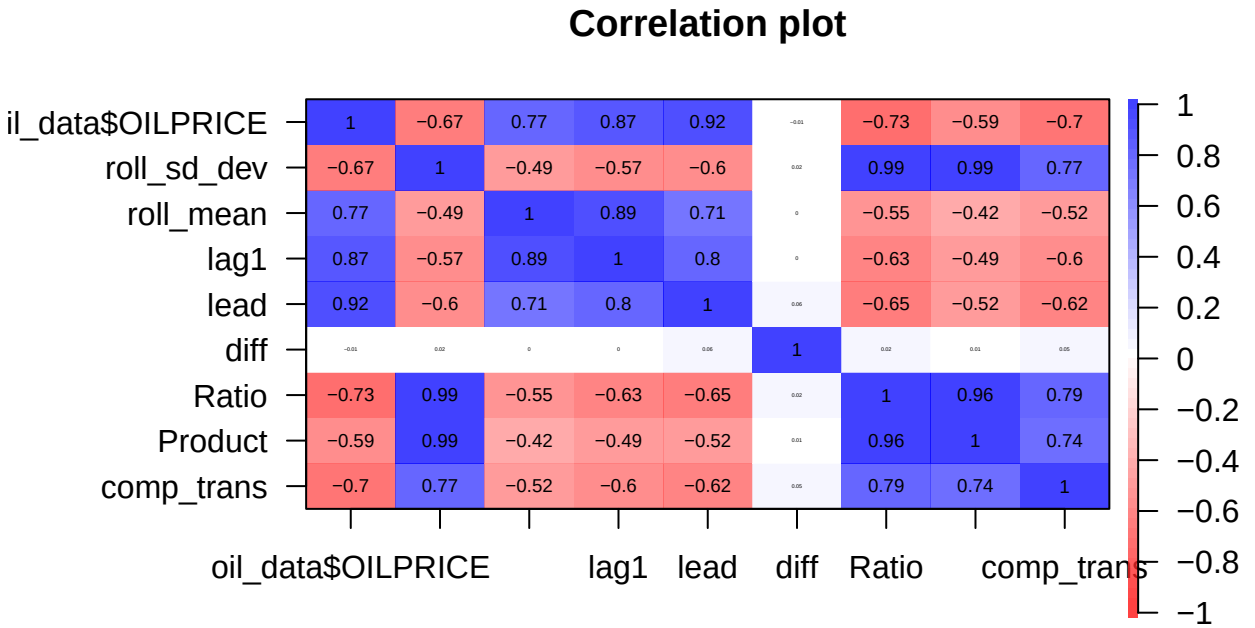

```
  shapiro.test(input_sd_dev)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  input_sd_dev
## W = 0.82788, p-value < 2.2e-16
```

```
# check stationarity
input_mean<- drivers$roll_mean
input_mean[is.na(input_mean)] <- 0
tseries::adf.test(input_mean)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  input_mean
## Dickey-Fuller = -1.1144, Lag order = 9, p-value = 0.9203
## alternative hypothesis: stationary
```

```
# check correlation
input<-drivers
input[is.na(input)] <- 0
psych::corPlot(input, cex = 0.4)
```



I have used following links for the references of this test. https://rdrr.io/cran/gamlss.data/man/oil.html https://www.kaggle.com/gabrieloliveirasan/gamlss-in-r-oil-price-prediction http://www.gamlss.com/wp-content/uploads/2013/01/gamlss-manual.pdf