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
Mydata <- function(rawdata) {
  oil_data <- rawdata
  return(oil_data)
}

oil_data <- Mydata(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
```

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

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

```r
Mypipeline1 <- function(rawdata1) {

df <- as.data.frame(rawdata1)

oil_data2 <- rawdata1
oil_data2 <- as.matrix(oil_data2)

# rtolling standard deviation with window = 5
roll_sdDev <- roll::roll_sd(oil_data2, 5)
df$roll_sdDev <- roll_sdDev

# Rolling mean with window = 5
roll_mean <- roll::roll_mean(oil_data2, 5)
df$roll_mean <- roll_mean

# Lagging with order = 2
df$lag1 <- dplyr::lag(rawdata1,2)

# Leading with order = 2
df$lead <- dplyr::lead(rawdata1,2)

Diff <- rawdata1 %>% diff()
Diff[1000] <- NA
df$diff <- Diff

return(df)
}

oil_trans <- Mypipeline1(oil_data$OILPRICE)

head(oil_trans, n =20)
```

```
##    rawdata1  roll_sdDev 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
## 11 4.649665 0.006262167  4.641255 4.640055 4.652721  0.0036280353
## 12 4.653293 0.007069724  4.644871 4.645544 4.664947 -0.0005719733
## 13 4.652721 0.005520886  4.648256 4.649665 4.656053  0.0122259022
## 14 4.664947 0.007234281  4.653234 4.653293 4.630253 -0.0088939937
## 15 4.656053 0.005832164  4.655336 4.652721 4.589955 -0.0258004299
## 16 4.630253 0.012822831  4.651454 4.664947 4.584355 -0.0402979462
## 17 4.589955 0.030141568  4.638786 4.656053 4.574814 -0.0055999739
## 18 4.584355 0.036972288  4.625113 4.630253 4.572750 -0.0095409803
## 19 4.574814 0.034591035  4.607086 4.589955 4.575535 -0.0020637712
## 20 4.572750 0.023342175  4.590425 4.584355 4.565701  0.0027850861
```

```
Mypipeline2 <- function(rawdata2){

  df1 <- as.data.frame(rawdata2)
  df1$Ratio <- df1[,1]/df1[,2]
  df1$Product <- df1[,1] * df1[,2]

  return(df1)

}
```

```
df1 <- Mypipeline2(oil_trans[, c("roll_sdDev", "roll_mean")])

head(df1, n = 20)
```

```
##     roll_sdDev 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
## 11 0.006262167  4.641255 0.001349240 0.02906432
## 12 0.007069724  4.644871 0.001522050 0.03283795
## 13 0.005520886  4.648256 0.001187733 0.02566249
## 14 0.007234281  4.653234 0.001554678 0.03366281
## 15 0.005832164  4.655336 0.001252791 0.02715068
## 16 0.012822831  4.651454 0.002756736 0.05964481
## 17 0.030141568  4.638786 0.006497727 0.13982028
## 18 0.036972288  4.625113 0.007993813 0.17100100
## 19 0.034591035  4.607086 0.007508224 0.15936387
## 20 0.023342175  4.590425 0.005084970 0.10715052
```

```
Mypipeline3 <- function(rawdata3){

  df2 <- as.data.frame(rawdata3)
  dif <- (df2$OILPRICE - df2$respLAG)
  df2$difference <- dif
  dif <- as.matrix(dif)
  roll_std <- roll::roll_sd(dif, 5)

  df2$comp_trans <- roll_std

  return(df2)

}

df2 <- Mypipeline3(oil_data[,c("OILPRICE", "respLAG")])

head(df2, n = 20)
```

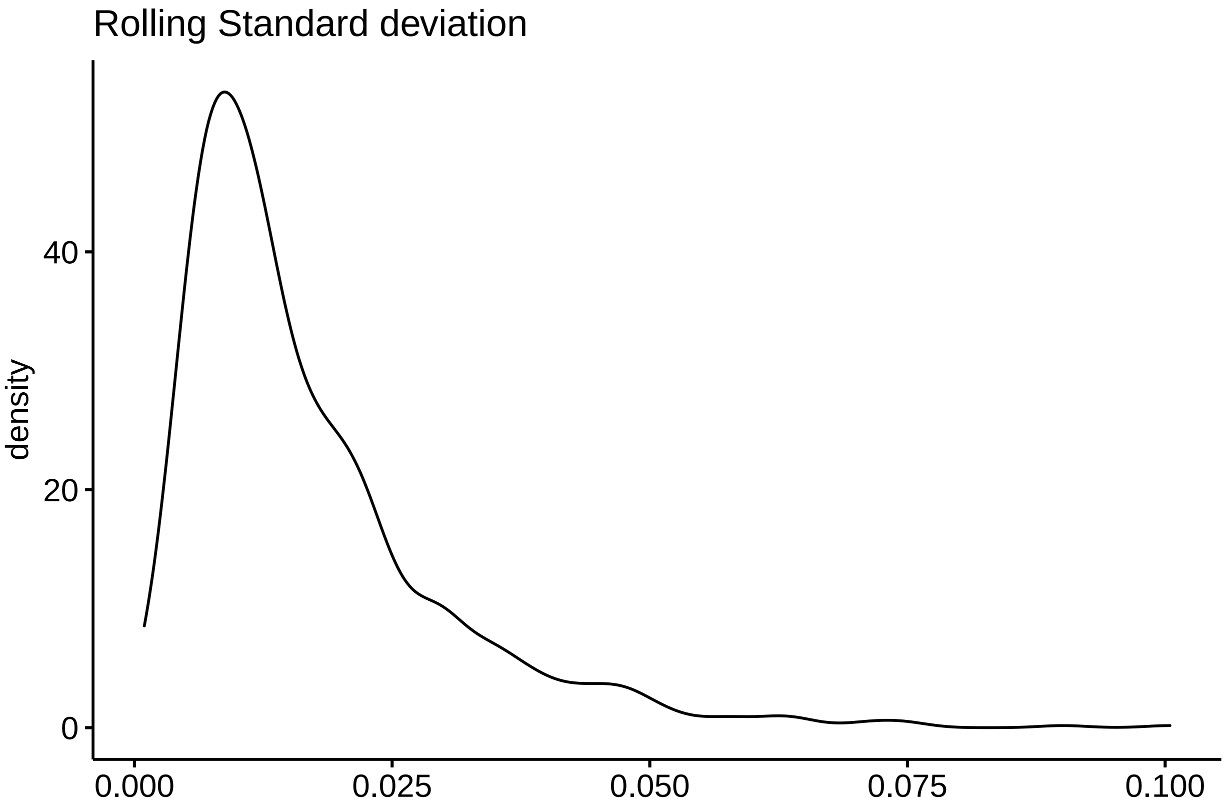
```
##    OILPRICE  respLAG    difference  comp_trans
## 1  4.640923 4.631812  0.0091112388          NA
## 2  4.633077 4.640923 -0.0078462165          NA
## 3  4.634049 4.633077  0.0009720063          NA
## 4  4.646312 4.634049  0.0122629838          NA
## 5  4.631520 4.646312 -0.0147921680 0.011343440
## 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
## 11 4.649665 4.645544  0.0041213457 0.002553802
## 12 4.653293 4.649665  0.0036280353 0.001829115
## 13 4.652721 4.653293 -0.0005719733 0.002315723
## 14 4.664947 4.652721  0.0122259022 0.004640910
## 15 4.656053 4.664947 -0.0088939937 0.007696784
## 16 4.630253 4.656053 -0.0258004299 0.014425317
## 17 4.589955 4.630253 -0.0402979462 0.020713148
## 18 4.584355 4.589955 -0.0055999739 0.020091840
## 19 4.574814 4.584355 -0.0095409803 0.014716288
## 20 4.572750 4.574814 -0.0020637712 0.016033983
```

```
drivers <- cbind(oil_trans, df1, df2)
drivers <-
  drivers[, c("rawdata1", "roll_sdDev", "roll_mean",
                       "lag1", "lead", "diff", "Ratio", "Product",
                       "comp_trans")]

head(drivers, n = 20)
```

```
##    rawdata1  roll_sdDev roll_mean     lag1     lead          diff       Ratio
## 1  4.640923          NA        NA       NA 4.634049 -0.0078462165          NA
## 2  4.633077          NA        NA       NA 4.646312  0.0009720063          NA
## 3  4.634049          NA        NA 4.640923 4.631520  0.0122629838          NA
## 4  4.646312          NA        NA 4.633077 4.627616 -0.0147921680          NA
## 5  4.631520 0.006246603  4.637176 4.634049 4.635214 -0.0039035865 0.001347070
## 6  4.627616 0.007035944  4.634515 4.646312 4.635796  0.0075979325 0.001518162
## 7  4.635214 0.006991536  4.634942 4.631520 4.640055  0.0005820722 0.001508441
## 8  4.635796 0.006979384  4.635292 4.627616 4.645544  0.0042582083 0.001505705
## 9  4.640055 0.004697145  4.634040 4.635214 4.649665  0.0054894923 0.001013618
## 10 4.645544 0.006612521  4.636845 4.635796 4.653293  0.0041213457 0.001426082
## 11 4.649665 0.006262167  4.641255 4.640055 4.652721  0.0036280353 0.001349240
## 12 4.653293 0.007069724  4.644871 4.645544 4.664947 -0.0005719733 0.001522050
## 13 4.652721 0.005520886  4.648256 4.649665 4.656053  0.0122259022 0.001187733
## 14 4.664947 0.007234281  4.653234 4.653293 4.630253 -0.0088939937 0.001554678
## 15 4.656053 0.005832164  4.655336 4.652721 4.589955 -0.0258004299 0.001252791
## 16 4.630253 0.012822831  4.651454 4.664947 4.584355 -0.0402979462 0.002756736
## 17 4.589955 0.030141568  4.638786 4.656053 4.574814 -0.0055999739 0.006497727
## 18 4.584355 0.036972288  4.625113 4.630253 4.572750 -0.0095409803 0.007993813
## 19 4.574814 0.034591035  4.607086 4.589955 4.575535 -0.0020637712 0.007508224
## 20 4.572750 0.023342175  4.590425 4.584355 4.565701  0.0027850861 0.005084970
##       Product  comp_trans
## 1          NA          NA
## 2          NA          NA
## 3          NA          NA
## 4          NA          NA
## 5  0.02896660 0.011343440
## 6  0.03260819 0.010142975
## 7  0.03240537 0.010514120
## 8  0.03235148 0.010510516
## 9  0.02176676 0.008695036
## 10 0.03066124 0.004534232
## 11 0.02906432 0.002553802
## 12 0.03283795 0.001829115
## 13 0.02566249 0.002315723
## 14 0.03366281 0.004640910
## 15 0.02715068 0.007696784
## 16 0.05964481 0.014425317
## 17 0.13982028 0.020713148
## 18 0.17100100 0.020091840
## 19 0.15936387 0.014716288
## 20 0.10715052 0.016033983
```

```
# check normality
normality_check <- function(input) {

  print(ggdensity(input,
          main = "Rolling Standard deviation",
          xlab = ""))

  shapiro.test(input)

}

normality_check(drivers$roll_sdDev)
```

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



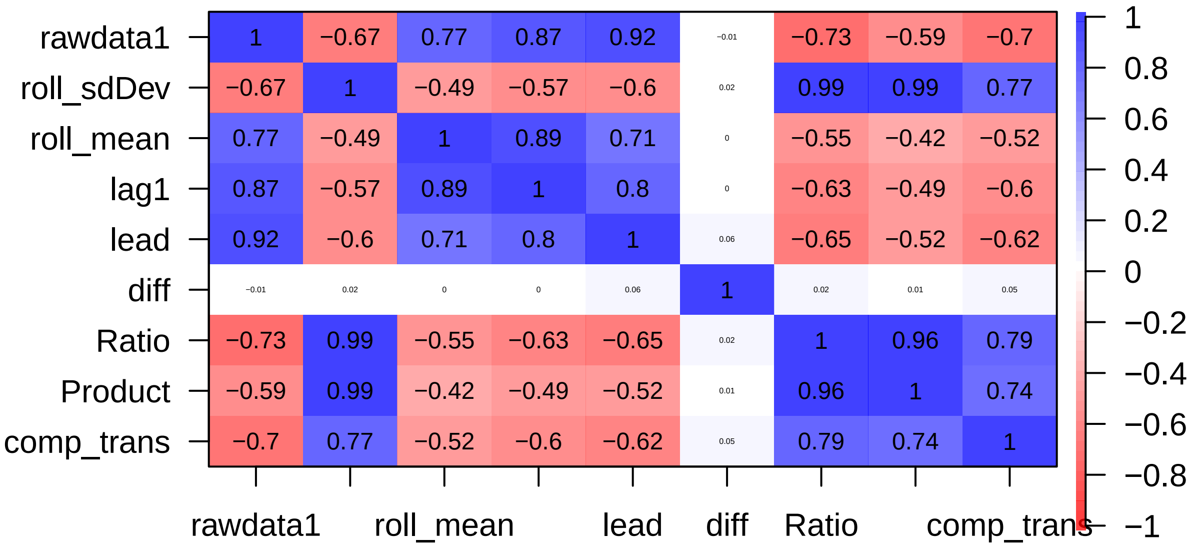
```
##
##  Shapiro-Wilk normality test
##
## data:  input
## W = 0.82788, p-value < 2.2e-16
```

```
# check stationarity
stationarity_chek <- function(input) {
  input[is.na(input)] <- 0
  tseries::adf.test(input)
}
stationarity_chek(drivers$roll_mean)
```

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

```
# check correlation
correlation_check <- function(input){

  input[is.na(input)] <- 0
  return(psych::corPlot(input, cex = 0.5))
}

correlation_check(drivers)
```

Correlation plot



	rawdata1		roll_mean		lead	diff	Ratio		comp_trans
rawdata1	1	−0.67	0.77	0.87	0.92	−0.01	−0.73	−0.59	−0.7
roll_sdDev	−0.67	1	−0.49	−0.57	−0.6	0.02	0.99	0.99	0.77
roll_mean	0.77	−0.49	1	0.89	0.71	0	−0.55	−0.42	−0.52
lag1	0.87	−0.57	0.89	1	0.8	0	−0.63	−0.49	−0.6
lead	0.92	−0.6	0.71	0.8	1	0.06	−0.65	−0.52	−0.62
diff	−0.01	0.02	0	0	0.06	1	0.02	0.01	0.05
Ratio	−0.73	0.99	−0.55	−0.63	−0.65	0.02	1	0.96	0.79
Product	−0.59	0.99	−0.42	−0.49	−0.52	0.01	0.96	1	0.74
comp_trans	−0.7	0.77	−0.52	−0.6	−0.62	0.05	0.79	0.74	1

```
# 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
```