Code

# Projet de serie temporelle Titre: Prévision de la demande d'electricité à court terme

**NOM de l'étudiante: KAMTCHUENG KOUOKAM Daline Anastasie**

INTRODUCTION_____

**Presentation de données**

Ce document présente une analyse de la consommation d'énergie sur une période de huit ans (2018 à 2025), réalisée à partir des données DOM_hourly disponibles sur Kaggle. Cette étude a été menée dans le cadre de mon évaluation de la Normale.

**Objectif** : Exploiter ces données afin de concevoir un modèle de prévision de la consommation d'énergie à court terme.

PARTIE I: Prétraitement de données_____

Importation des données

Hide

```
data <- read.csv("DOM_hourly.csv")
head(data)
```

| | Datetime<br><chr> | DOM_MW<br><dbl> |
|---|---|---|
| 1 | 2005-12-31 01:00:00 | 9389 |
| 2 | 2005-12-31 02:00:00 | 9070 |
| 3 | 2005-12-31 03:00:00 | 9001 |
| 4 | 2005-12-31 04:00:00 | 9042 |
| 5 | 2005-12-31 05:00:00 | 9132 |
| 6 | 2005-12-31 06:00:00 | 9353 |

6 rows

Nous allons nous assurer ensuite que la colonne DOM_MW contient bien des valeurs numerique et suprimer les valeur manquante

Hide

```
#convertion de DOM_MW envaleur numerique
data$DOM_MW = as.numeric(data$DOM_MW)

#Elimination des valeurs manquantes
data$DOM_MW[is.na(data$DOM_MW)] = 0

#convertion de la colone datatiome
data$Datetime <- as.Date(data$Datetime)
```

Différentes librairies utilisées:

Hide

```
library(TSstudio)
library(tidyverse)
library(dplyr)
library(lubridate)
library(dygraphs)
library(forecast)
library(zoo)
library(xts)
```

Après observation de nos données, nous constatons qu'elles varient sur 23 minutes chaque jour la plupart du temps, mais de façon aléatoire, car nous avons observé des journées avec une seule observation. Pour résoudre ce problème, nous allons effectuer la moyenne des données pour les minutes de chaque journée afin d'obtenir des données journalières.

Hide

```
data_jour <- data %>%
  mutate(date = as.Date(Datetime)) %>%
  group_by(date) %>%
  summarise(DOM_MW = mean(DOM_MW, na.rm = TRUE))
```

Nous obtenons donc des données qui ne dépendent plus du temps et ont une valeur journalière.

Hide

```
head(data_jour)
```

| | date<br><date> | DOM_MW<br><dbl> |
|---|---|---|
| | 2005-05-01 | 7812.348 |
| | 2005-05-02 | 8608.083 |
| | 2005-05-03 | 8665.000 |
| | 2005-05-04 | 8628.792 |
| | 2005-05-05 | 8702.542 |
| | 2005-05-06 | 9008.542 |

6 rows

PARTIE II: Etude de notre modéle

---

Nous allons maintenant effectuer la transformation de nos données en données de série temporelle.

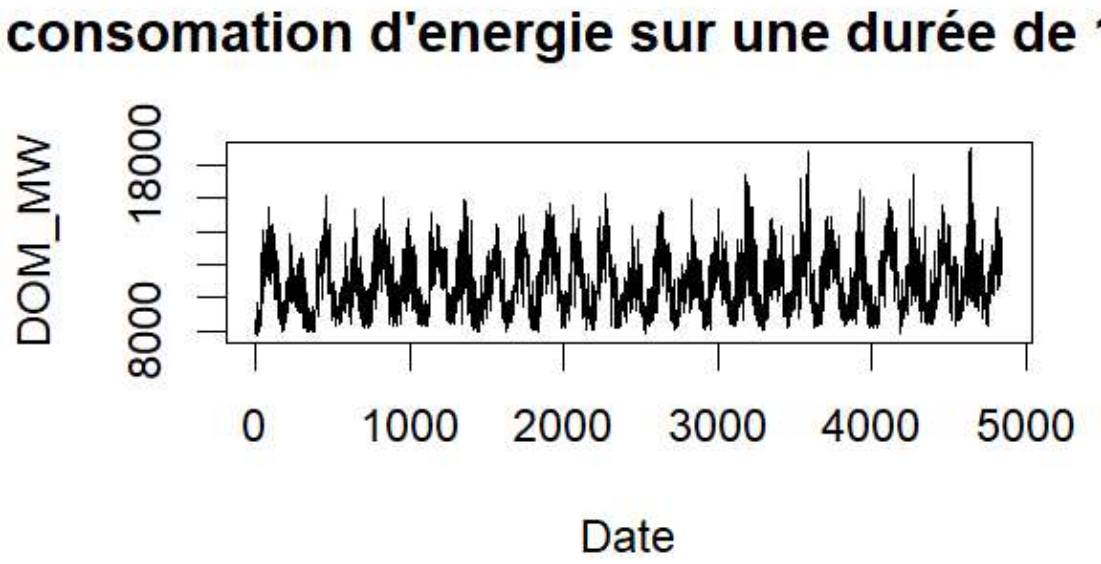
Hide

```
data_ts <- xts(data_jour$DOM_MW, order.by = data_jour$date)
```

Visualisation de nos donner de serie temprelle

Hide

```
plot.ts(
  data_ts,
  main="consomation d'energie sur une durée de 14ans",
  xlab="Date",
  ylab="DOM_MW"
)
```




Nous allons maintenant effectuer la subdivision de nos données en données d'entraînement et données de test, afin de pouvoir entraîner notre modèle et réaliser des prévisions.

Hide

```
# Définir la taille de l'ensemble d'entraînement
train_size <- floor(length(data_ts) * 0.7)

# Partitionner les données en ensemble d'entraînement et ensemble de test
train = window(data_ts, end = index(data_ts)[train_size])


test = window(data_ts, start = index(data_ts)[train_size + 1])
```

Informations sur nos données d'entraînement :

Hide

```
ts_info(train)
```

```
 The train series is a xts object with 1 variable and 3390 observations
 Frequency: daily
 Start time: 2005-05-01
 End time: 2014-08-11
```

Hide

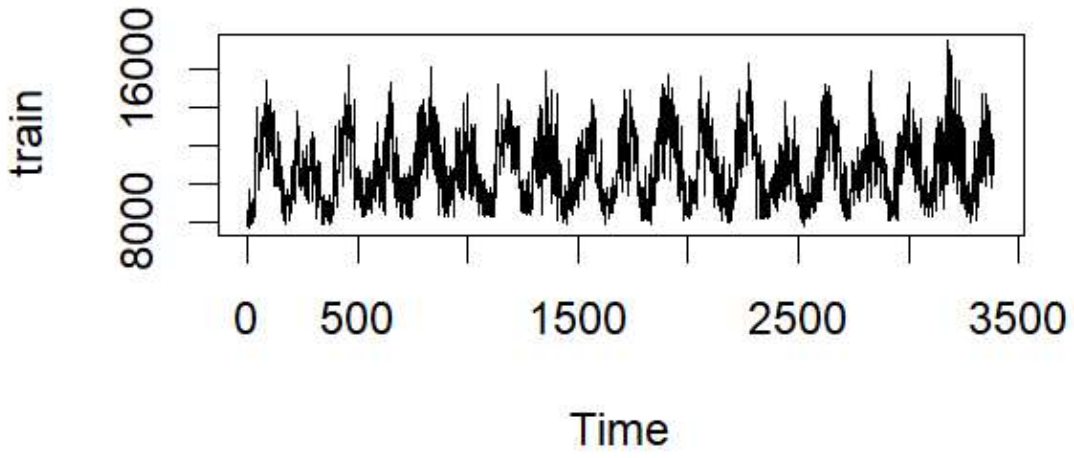
```
head(train)
```

```
             [,1]
2005-05-01 7812.348
2005-05-02 8608.083
2005-05-03 8665.000
2005-05-04 8628.792
2005-05-05 8702.542
2005-05-06 9008.542
```

Visualisation des données d'entraînement:

Hide

```
plot.ts(train)
```




Informations sur nos données de test :

Hide

```
ts_info(test)
```

```
 The test series is a xts object with 1 variable and 1453 observations
 Frequency: daily
 Start time: 2014-08-12
 End time: 2018-08-03
```

Hide

```
head(test)
```

```
             [,1]
2014-08-12 11695.42
2014-08-13 11989.42
2014-08-14 11081.08
2014-08-15 10800.88
2014-08-16 10772.21
2014-08-17 11085.79
```

Entraînement de notre modèle avec Auto ARIMA :

Hide

```
model <- auto.arima(train)
```

Hide

```
print(model)
```

```
Series: train
ARIMA(5,0,1) with non-zero mean

Coefficients:
         ar1      ar2     ar3      ar4     ar5      ma1        mean
      1.8004  -1.3085  0.6444  -0.2886  0.1342  -0.7005  10869.2179
s.e.  0.0534   0.0651  0.0449   0.0350  0.0214   0.0528    210.5045

sigma^2 = 564218:  log likelihood = -27254.98
AIC=54525.96   AICc=54526   BIC=54574.98
```

Visualisation des résidus du modèle ainsi que des autocorrélations (ACF) :

Hide

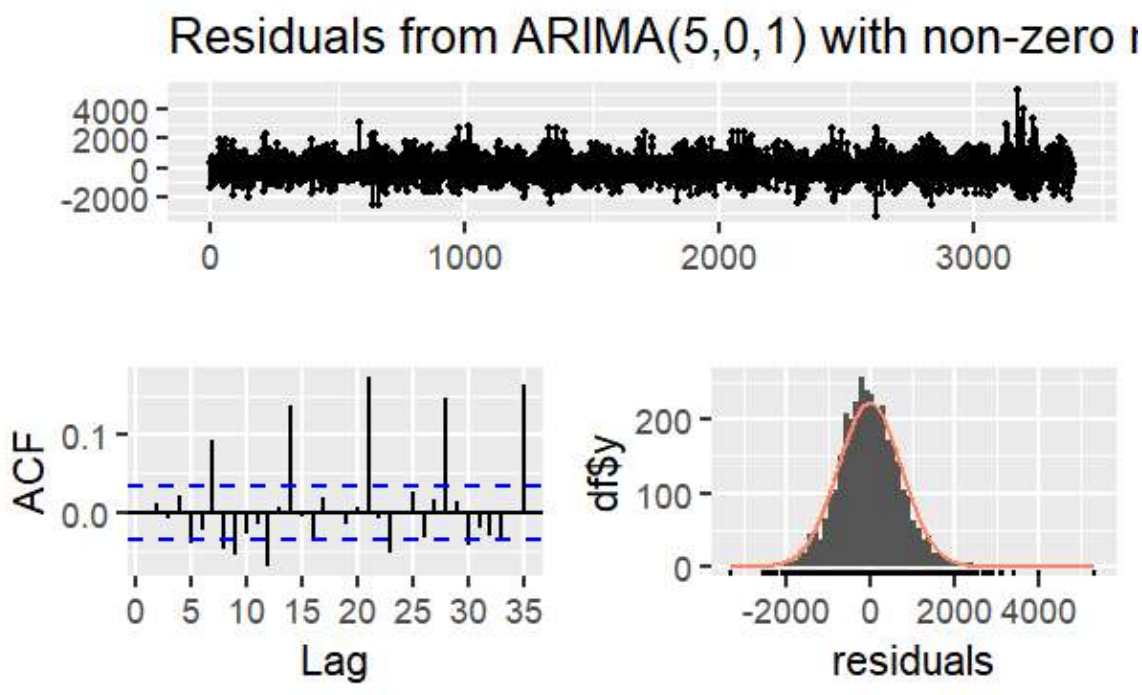
```
checkresiduals(model)
```

```

    Ljung-Box test

data:  Residuals from ARIMA(5,0,1) with non-zero mean
Q* = 58.707, df = 4, p-value = 5.422e-12

Model df: 6.   Total lags used: 10
```




Effectuons maintenant les prédictions avec notre modèle sur la même période que les données de test :

Hide

```
fc=forecast(model,length(test))
print(fc)
```

| | Point Forecast <dbl> | Lo 80 <dbl> | Hi 80 <dbl> | Lo 95 <dbl> | Hi 95 <dbl> |
|---|---|---|---|---|---|
| 3391 | 11352.45 | 10389.821 | 12315.08 | 9880.235 | 12824.67 |
| 3392 | 11370.35 | 9939.397 | 12801.31 | 9181.895 | 13558.81 |
| 3393 | 11339.47 | 9769.230 | 12909.71 | 8937.995 | 13740.95 |
| 3394 | 11258.30 | 9638.172 | 12878.43 | 8780.527 | 13736.07 |
| 3395 | 11184.30 | 9540.713 | 12827.88 | 8670.651 | 13697.95 |
| 3396 | 11150.60 | 9492.839 | 12808.37 | 8615.272 | 13685.93 |
| 3397 | 11145.78 | 9470.729 | 12820.82 | 8584.013 | 13707.54 |
| 3398 | 11152.77 | 9451.265 | 12854.28 | 8550.541 | 13755.00 |
| 3399 | 11160.44 | 9425.841 | 12895.04 | 8507.600 | 13813.28 |
| 3400 | 11161.77 | 9393.139 | 12930.40 | 8456.882 | 13866.66 |

1-10 of 1,453 rows Previous **1** 2 3 4 5 6 … 100 Next

Maintenant, estimons l'erreur entre les données prédites et les données de test :

Hide

```
accuracy(fc,test)
```

```
                    ME       RMSE       MAE         MPE      MAPE      MASE          ACF1
Training set   0.8821897  750.3682  576.5216 -0.4458901  5.292597 0.8996328 -0.002346468
Test set     274.2535854 1896.3786 1515.1259 -0.1337928 13.307374 2.3642771            NA
```

Effectuons maintenant des prévisions sur la consommation d'énergie à partir de nos données pour une période de 10 jours :

Hide

```
new_model=auto.arima(data_ts)
prevision_journaliére=forecast(new_model,h=20)
print(prevision_journaliére)
```

| | Point Forecast <dbl> | Lo 80 <dbl> | Hi 80 <dbl> | Lo 95 <dbl> | Hi 95 <dbl> |
|---|---|---|---|---|---|
| 4844 | 11159.96 | 10162.775 | 12157.14 | 9634.898 | 12685.02 |
| 4845 | 11607.97 | 10119.739 | 13096.21 | 9331.915 | 13884.04 |
| 4846 | 11839.46 | 10202.713 | 13476.22 | 9336.269 | 14342.66 |
| 4847 | 11888.22 | 10198.837 | 13577.60 | 9304.531 | 14471.91 |
| 4848 | 11866.40 | 10143.286 | 13589.51 | 9231.126 | 14501.67 |
| 4849 | 11859.23 | 10104.661 | 13613.80 | 9175.849 | 14542.61 |

| | Point Forecast <dbl> | Lo 80 <dbl> | Hi 80 <dbl> | Lo 95 <dbl> | Hi 95 <dbl> |
|---|---|---|---|---|---|
| 4850 | 11827.31 | 10045.772 | 13608.85 | 9102.681 | 14551.94 |
| 4851 | 11817.46 | 10008.838 | 13626.09 | 9051.410 | 14583.51 |
| 4852 | 11786.94 | 9954.584 | 13619.29 | 8984.594 | 14589.28 |
| 4853 | 11776.99 | 9920.739 | 13633.24 | 8938.099 | 14615.88 |

1-10 of 20 rows Previous 1 2 Next

Visualisons maintenant nos données de prévisions:

Hide

```
test_forecast(
  actual = data_ts,
  forecast.obj = fc,
  test=test
)
```