Titre : Analyse sous R de la production mensuelle d'électricité en Tunisie (2005–2024)

- Ines elouaer

1) Introduction

- Cette étude analyse l'évolution mensuelle de la production d'électricité en Tunisie entre 2005 et 2024, ventilée par type de producteur (STEG, IPP, auto-producteurs et solaire). L'objectif est de décrire les tendances, la saisonnalité et les différences entre producteurs, ainsi que de proposer un modèle statistique simple et interprétable permettant d'expliquer la production totale.

> 1.1 Objectifs

Les objectifs de l'analyse sont :

- Décrire la production totale et son évolution dans le temps
- Comparer les producteurs (niveaux, dispersion, apparition du solaire/auto)
- Mettre en évidence la saisonnalité mensuelle
- Tester si la production STEG diffère significativement de celle des IPP
- Modéliser la production totale et choisir un modèle pertinent

2) Import + inspection initiale

```
library(readr)
library(dplyr)
library(ggplot2)
library(lubridate)
```

```
data <- read.csv(
  "sample_data/opendataproductionmensuelleelectriciteparproducteurs.csv"
  sep = ";",
  stringsAsFactors = FALSE
)
```

2.1. Compréhension initiale des données

```
str(data)
summary(data)
head(data, 10)
```

```
'data.frame':   236 obs. of  6 variables:
 $ Date                                : chr  "1 / 2005" "2 / 2005" "3 / 2005"
 $ Production.STEG..en.GWh             : num  715 668 669 764 701 ...
 $ Production.IPP.en.GWh               : num  254 242 270 120 269 ...
 $ IPP.Solaire                         : num  NA NA NA NA NA NA NA NA NA NA ..
 $ Production.Auto.producteurs..en.GWh: num  NA NA NA NA NA NA NA NA NA NA ..
 $ Production.totale..GWh.             : num  969 910 939 884 971 ...
    Date           Production.STEG..en.GWh Production.IPP.en.GWh
Length:236         Min.   : 648.2          Min.   :  0.0
Class :character   1st Qu.: 917.1          1st Qu.:243.6
Mode  :character   Median :1136.1          Median :276.3
                   Mean   :1171.9          Mean   :235.6
                   3rd Qu.:1328.4          3rd Qu.:293.1
                   Max.   :2492.6          Max.   :321.4

 IPP.Solaire    Production.Auto.producteurs..en.GWh Production.totale..GWh.
Min.   :1.250   Min.   : 2.324                      Min.   : 883.9
1st Qu.:1.950   1st Qu.:24.388                      1st Qu.:1181.7
Median :3.200   Median :33.946                      Median :1380.5
Mean   :2.970   Mean   :34.019                      Mean   :1410.6
3rd Qu.:3.725   3rd Qu.:44.795                      3rd Qu.:1535.4
Max.   :4.840   Max.   :64.649                      Max.   :2541.9
NA's   :216     NA's   :216
```

A data.frame: 10 × 6

	Date	Production.STEG..en.GWh	Production.IPP.en.GWh	IPP.Solaire	Produ
	<chr>	<dbl>	<dbl>	<dbl>	
1	1 / 2005	715.18	253.95	NA	
2	2 / 2005	667.95	241.94	NA	
3	3 / 2005	669.34	269.98	NA	
4	4 / 2005	764.08	119.77	NA	
5	5 / 2005	701.34	269.35	NA	
6	6 / 2005	885.49	157.03	NA	
7	7 / 2005	946.21	278.99	NA	
8	8 / 2005	892.76	283.16	NA	
9	9 / 2005	791.34	257.46	NA	
10	10 / 2005	712.34	270.78	NA	

3) Nettoyage : renommage des variables

- Les noms de colonnes sont renommés pour améliorer la lisibilité du code et faciliter l'analyse. Cette étape réduit les erreurs et rend les graphiques/modèles plus clairs.

```
data <- data %>%
  rename(
    date = Date,
    steg = Production.STEG..en.GWh,
    ipp = Production.IPP.en.GWh,
    solaire = IPP.Solaire,
    auto = Production.Auto.producteurs..en.GWh,
    total = Production.totale..GWh.
  )

```

```
str(data)
summary(data)

```

```
'data.frame':   236 obs. of  6 variables:
 $ date   : chr  "1 / 2005" "2 / 2005" "3 / 2005" "4 / 2005" ...
 $ steg   : num  715 668 669 764 701 ...
 $ ipp    : num  254 242 270 120 269 ...
 $ solaire: num  NA NA NA NA NA NA NA NA NA NA ...
 $ auto   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ total  : num  969 910 939 884 971 ...
    date                steg             ipp           solaire
 Length:236         Min.   : 648.2   Min.   :  0.0   Min.   :1.250
 Class :character   1st Qu.: 917.1   1st Qu.:243.6   1st Qu.:1.950
 Mode  :character   Median :1136.1   Median :276.3   Median :3.200
                    Mean   :1171.9   Mean   :235.6   Mean   :2.970
                    3rd Qu.:1328.4   3rd Qu.:293.1   3rd Qu.:3.725
                    Max.   :2492.6   Max.   :321.4   Max.   :4.840
                                                     NA's   :216
     auto            total
 Min.   : 2.324   Min.   : 883.9
 1st Qu.:24.388   1st Qu.:1181.7
 Median :33.946   Median :1380.5
 Mean   :34.019   Mean   :1410.6
 3rd Qu.:44.795   3rd Qu.:1535.4
 Max.   :64.649   Max.   :2541.9
 NA's   :216
```

3.2 Nettoyage : conversion des dates (formats mixtes FR)

- La variable date est fournie sous plusieurs formats (ex. m / Y et janv-22). Une conversion en véritable format date est nécessaire pour réaliser des séries temporelles correctes et des analyses saisonnières.

```r
library(stringr)

to_date_fr <- function(x){
  x <- str_trim(x)

  # Cas 1 : "1 / 2005" (mois / année)
  # on enlève les espaces et on parse
  is_num <- str_detect(x, "^\\d+\\s*/\\s*\\d{4}$")

  out <- rep(as.Date(NA), length(x))

  # Parser "m / Y"
  out[is_num] <- my(str_replace_all(x[is_num], "\\s+", ""))  # my = mont

  # Cas 2 : "janv-22", "févr-22", etc.
  # On remplace mois FR -> EN abréviations reconnues (Jan, Feb, Mar...)
  m <- x[!is_num]

  # Corriger encodage fréquent: "f�vr", "d�c", "ao�t" (caractères cass
  m <- str_replace_all(m, c("f�vr"="févr","d�c"="déc","ao�t"="août"))

  # map mois FR -> EN
  m2 <- str_replace_all(m, c(
    "janv"="Jan", "févr"="Feb", "fevr"="Feb", "mars"="Mar", "avr"="Apr",
    "mai"="May", "juin"="Jun", "juil"="Jul", "août"="Aug", "aout"="Aug",
    "sept"="Sep", "oct"="Oct", "nov"="Nov", "déc"="Dec", "dec"="Dec"
  ))

  # parser "Mon-YY"
  out[!is_num] <- my(m2)

  out
}

```

```r
data$date <- to_date_fr(data$date)
```

```r
head(data$date, 15)
tail(data$date, 15)
range(data$date, na.rm = TRUE)

```

```
2005-01-01 · 2005-02-01 · 2005-03-01 · 2005-04-01 · 2005-05-01 · 2005-06-01 · 2005-07-01 · 2005-08-01 · 2005-09-01 · 2005-10-01 · 2005-11-01 · 2005-12-01 · 2006-01-01 · 2006-02-01 · 2006-03-01
2023-06-01 · 2023-07-01 · 2023-08-01 · 2023-09-01 · 2023-10-01 · 2023-11-01 · 2023-12-01 · 2024-01-01 · 2024-02-01 · 2024-03-01 · 2024-04-01 · 2024-05-01 · 2024-06-01 · 2024-07-01 · 2024-08-01
2005-01-01 · 2024-08-01
```

4) Contrôle qualité : NA + doublons

- Gestion des valeurs manquantes (NA) : La vérification des valeurs manquantes montre que les NA concernent principalement solaire et auto, disponibles uniquement sur les années récentes. Ces NA ne sont pas supprimés ni remplacés afin de préserver la cohérence temporelle. Pour les statistiques descriptives et certaines visualisations, les calculs sont réalisés avec na.rm=TRUE. Pour les analyses dédiées à la production solaire ou aux auto-producteurs, un filtrage ciblé (filter(!is.na(...))) est appliqué afin de se limiter à la période où ces variables existent.

```
colSums(is.na(data))
summary(is.na(data))
```

```
date:       0 steg:       0 ipp:       0 solaire:       216 auto:       216 total:       0
    date            steg            ipp            solaire
 Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:236       FALSE:236       FALSE:236       FALSE:20
                                                 TRUE :216
    auto            total
 Mode :logical   Mode :logical
 FALSE:20        FALSE:236
 TRUE :216
```

```
sum(duplicated(data))
data %>% filter(duplicated(.))

```

```
0
```

A data.frame: 0 × 6

date	steg	ipp	solaire	auto	total
<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>

5) Transformation (Pivot wide -> long)

- La transformation en format long (tidy data) permet de regrouper les producteurs dans une seule colonne (producteur) et leurs valeurs dans une autre (production), ce qui simplifie les tableaux de synthèse et les visualisations comparatives.

```r
library(tidyr)

data_long <- data %>%
  pivot_longer(
    cols = c(steg, ipp, solaire, auto),
    names_to = "producteur",
    values_to = "production"
  )

```

```r
str(data_long)
summary(data_long)

```

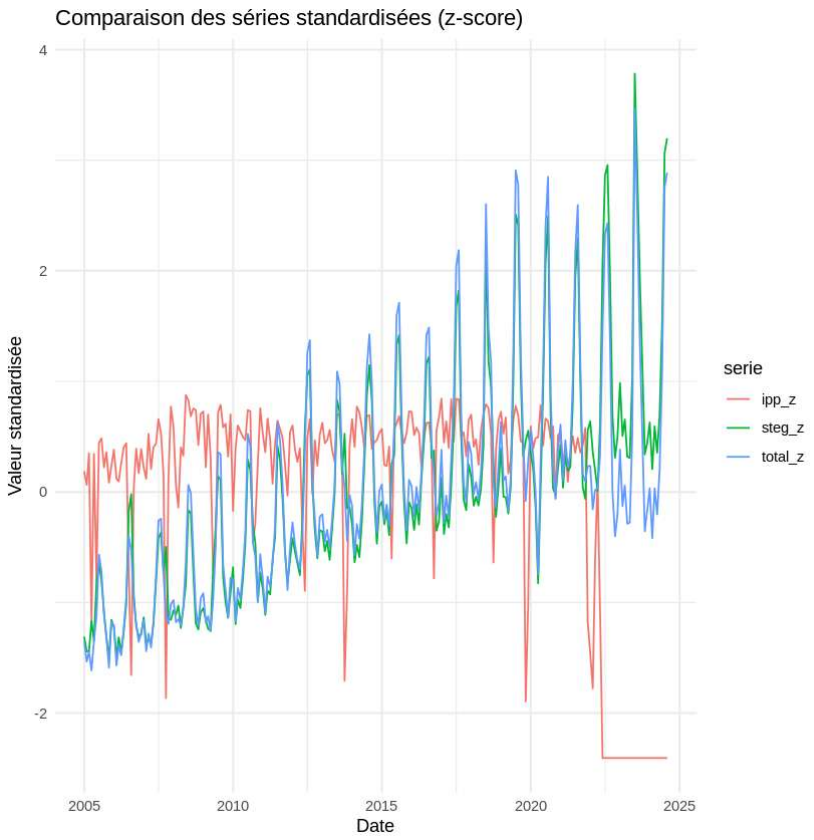
```
tibble [944 × 4] (S3: tbl_df/tbl/data.frame)
 $ date      : Date[1:944], format: "2005-01-01" "2005-01-01" ...
 $ total     : num [1:944] 969 969 969 969 910 ...
 $ producteur: chr [1:944] "steg" "ipp" "solaire" "auto" ...
 $ production: num [1:944] 715 254 NA NA 668 ...
      date                total          producteur           production     
 Min.   :2005-01-01   Min.   : 883.9   Length:944         Min.   :   0.0  
 1st Qu.:2009-11-23   1st Qu.:1181.7   Class :character   1st Qu.: 265.2  
 Median :2014-10-16   Median :1380.5   Mode  :character   Median : 306.8  
 Mean   :2014-10-16   Mean   :1410.6                      Mean   : 650.2  
 3rd Qu.:2019-09-08   3rd Qu.:1535.4                      3rd Qu.:1114.2  
 Max.   :2024-08-01   Max.   :2541.9                      Max.   :2492.6  
                                                          NA's   :432     
```

6) Normalisation

- La normalisation n’est pas nécessaire pour l’analyse en GWh, mais une standardisation (z-score) peut être utilisée pour comparer visuellement les dynamiques sur une même échelle.

```r
data_scaled <- data %>%
  mutate(
    steg_z  = as.numeric(scale(steg)),
    ipp_z   = as.numeric(scale(ipp)),
    total_z = as.numeric(scale(total))
  )

data_scaled_long <- data_scaled %>%
  select(date, steg_z, ipp_z, total_z) %>%
  pivot_longer(-date, names_to = "serie", values_to = "z")

ggplot(data_scaled_long, aes(x = date, y = z, color = serie)) +
  geom_line() +
  theme_minimal() +
  labs(
    title = "Comparaison des séries standardisées (z-score)",
    x = "Date",
    y = "Valeur standardisée"

```

```
19  )
20
```



7) Statistiques descriptives (tableaux de synthèse)

- Nous calculons des statistiques descriptives par producteur. Le nombre d'observations n montre que solaire/auto sont disponibles seulement sur les années récentes.

```
tab_stats <- data_long %>%
  group_by(producteur) %>%
  summarise(
    n = sum(!is.na(production)),
    moyenne = mean(production, na.rm = TRUE),
    mediane = median(production, na.rm = TRUE),
    ecart_type = sd(production, na.rm = TRUE),
    min = min(production, na.rm = TRUE),
    max = max(production, na.rm = TRUE)
  )

tab_stats
```

A tibble: 4 × 7

producteur	n	moyenne	mediane	ecart_type	min	max
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
auto	20	34.01944	33.94566	19.09729	2.323957	64.64851
ipp	236	235.64076	276.31500	97.91540	0.000000	321.42000
solaire	20	2.97000	3.20000	1.04400	1.250000	4.84000
steg	236	1171.86673	1136.06000	349.08526	648.180000	2492.59000

8) Analyse exploratoire (EDA) : graphiques

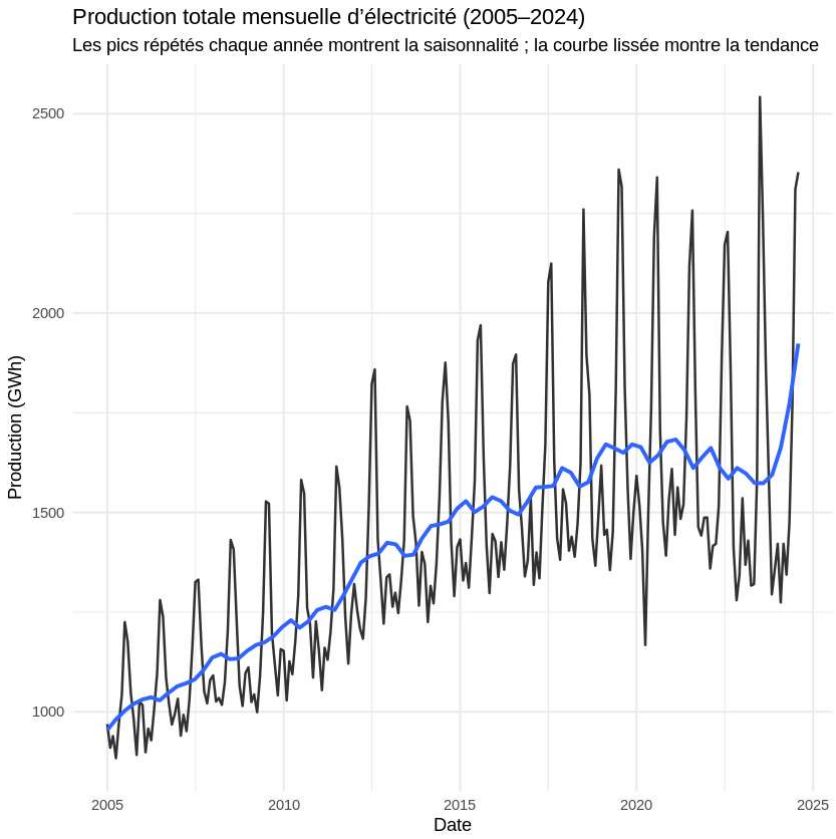
- L'analyse exploratoire (EDA) permet d'identifier visuellement la tendance globale, la saisonnalité et les différences de niveaux entre producteurs avant de passer aux tests statistiques et aux modèles.

Figure 1 — production totale (tendance + saisonnalité):

Cette figure représente la production totale d'électricité chaque mois entre 2005 et 2024. La courbe noire montre les variations mensuelles, avec des pics qui reviennent régulièrement, ce qui indique une saisonnalité. La courbe bleue (lissée) résume la tendance globale : la production augmente au fil du temps, même si certains mois restent plus élevés que d'autres.

```r
library(dplyr)
library(lubridate)
library(ggplot2)

data_plot <- data %>%
  arrange(date) %>%
  mutate(
    year  = year(date),
    month = month(date)
  )

ggplot(data_plot, aes(x = date, y = total)) +
  geom_line(linewidth = 0.7, alpha = 0.8) +
  geom_smooth(method = "loess", span = 0.15, se = FALSE, linewidth = 1.1
  labs(
    title = "Production totale mensuelle d'électricité (2005-2024)",
    subtitle = "Les pics répétés chaque année montrent la saisonnalité ;
    x = "Date",
    y = "Production (GWh)"
  ) +
  theme_minimal()

```

```
`geom_smooth()` using formula = 'y ~ x'
```



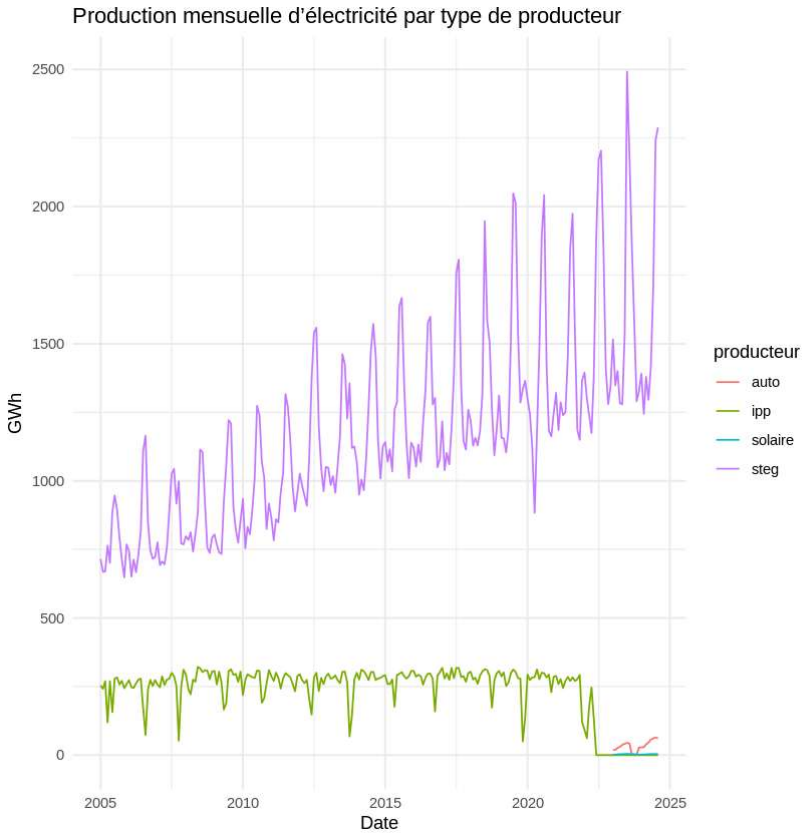
2. Figure 2 – Production mensuelle d'électricité par type de producteur

---

Ce graphique illustre l'évolution mensuelle de la production d'électricité en Tunisie selon le type de producteur. La STEG reste le principal producteur avec une tendance croissante et une forte saisonnalité. Les IPP jouent un rôle complémentaire et plus stable, tandis que la production solaire et celle des auto-producteurs apparaissent récemment et demeurent encore marginales.

```
# 7.2 Série temporelle par producteur (NA ignorés automatiquement)
ggplot(data_long, aes(x = date, y = production, color = producteur)) +
  geom_line() +
  labs(
    title = "Production mensuelle d'électricité par type de producteur",
    x = "Date", y = "GWh"
  ) +
  theme_minimal()
```

```
Warning message:
“Removed 432 rows containing missing values or values outside the scale range
(`geom_line()`).”
```

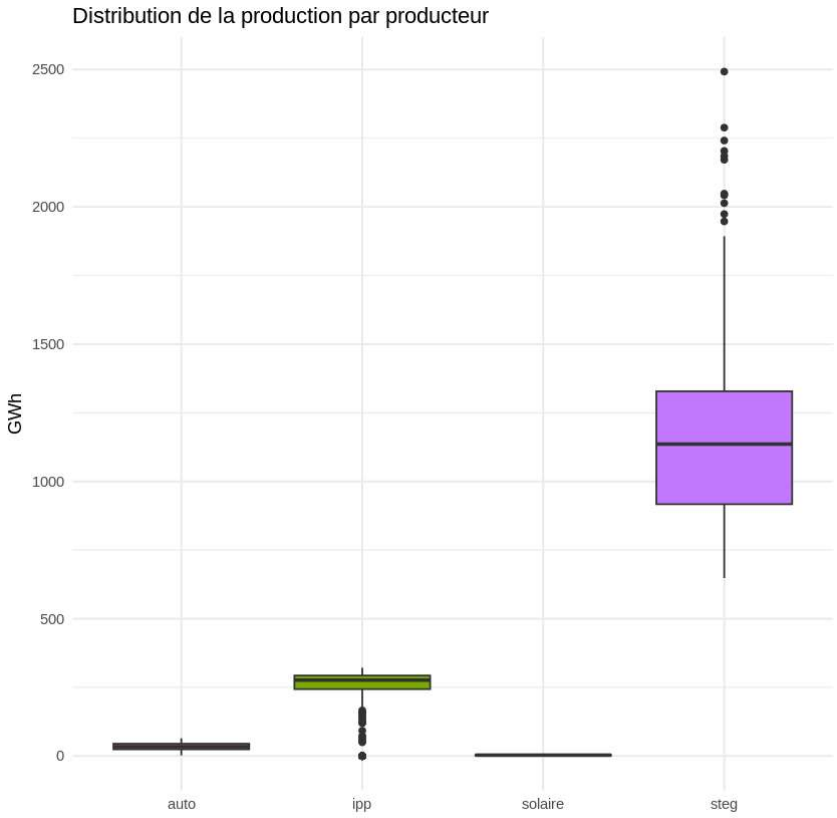


3. Figure 3 – Boxplot : Distribution de la production par producteur

---

Ce boxplot permet de comparer rapidement les niveaux et la variabilité de la production selon le type de producteur. Il montre que la STEG produit nettement plus que les autres et présente une forte dispersion (variations mensuelles importantes). Les points extrêmes correspondent à des mois de production très élevée. Les IPP ont une production plus faible et plus stable, tandis que le solaire et les auto-producteurs restent encore très marginaux.

```
# 7.3 Boxplot par producteur (comparaison des distributions)
ggplot(data_long, aes(x = producteur, y = production, fill = producteur)
  geom_boxplot(na.rm = TRUE) +
  labs(title = "Distribution de la production par producteur", x = "", y
  theme_minimal() +
  theme(legend.position = "none")
```



9) Préparer les données (steg & ipp sans NA)

```
df_test <- data %>% select(steg, ipp) %>% na.omit()
```

9.1 Test de normalité (Shapiro) sur chaque variable

- Nous utilisons un test t apparié car les deux séries sont observées sur les mêmes mois : Les tests de normalité montrent que les productions mensuelles de la STEG et des IPP ne suivent pas une distribution normale. Cependant, comme le nombre d'observations est élevé, le test t apparié reste valable. Les résultats indiquent une différence très significative entre les deux producteurs, la STEG produisant en moyenne environ 936 GWh de plus par mois que les IPP.

```
# Normalité (indicatif)
shapiro.test(df_test$steg)
shapiro.test(df_test$ipp)

# Test t apparié
t_test_res <- t.test(df_test$steg, df_test$ipp, paired = TRUE)
t_test_res

```

```
	Shapiro-Wilk normality test

data:  df_test$steg
W = 0.92947, p-value = 3.402e-09

	Shapiro-Wilk normality test

data:  df_test$ipp
W = 0.6816, p-value < 2.2e-16

	Paired t-test

data:  df_test$steg and df_test$ipp
t = 36.15, df = 235, p-value < 2.2e-16
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 885.2031 987.2488
sample estimates:
mean difference
        936.226
```

10) régression

- Nous comparons plusieurs modèles : (1) tendance seule, (2) tendance + saisonnalité. Le modèle temps+saison est retenu car il est interprétable et explique mieux la variabilité.

```
data <- data %>%
  arrange(date) %>%
  mutate(
    time = row_number(),
    month = month(date)
  )

m1 <- lm(total ~ time, data = data)
m2 <- lm(total ~ time + factor(month), data = data)

summary(m1)
summary(m2)

data.frame(
  modele = c("Temps", "Temps + Saison"),
  R2 = c(summary(m1)$r.squared, summary(m2)$r.squared),
  R2_ajuste = c(summary(m1)$adj.r.squared, summary(m2)$adj.r.squared)
)

```

```
Call:
lm(formula = total ~ time, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-491.12 -141.05  -67.41   82.56  798.71 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1033.5150    31.8765   32.42   <2e-16 ***
time           3.1825     0.2332   13.65   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 244.1 on 234 degrees of freedom
Multiple R-squared:  0.4432,	Adjusted R-squared:  0.4408 
F-statistic: 186.2 on 1 and 234 DF,  p-value: < 2.2e-16

Call:
lm(formula = total ~ time + factor(month), data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-309.94  -41.96   17.48   64.22  363.88 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      978.1524    29.0069  33.721  < 2e-16 ***
time               3.1227     0.1113  28.061  < 2e-16 ***
factor(month)2  -119.2110    36.8140  -3.238 0.001386 ** 
factor(month)3   -72.9494    36.8145  -1.982 0.048759 *  
factor(month)4  -137.1369    36.8153  -3.725 0.000247 ***
factor(month)5   -39.8172    36.8165  -1.082 0.280641    
factor(month)6   141.8336    36.8180   3.852 0.000153 ***
factor(month)7   503.5419    36.8198  13.676  < 2e-16 ***
factor(month)8   482.7278    36.8220  13.110  < 2e-16 ***
factor(month)9   168.0198    37.2957   4.505 1.07e-05 ***
factor(month)10  -23.7770    37.2965  -0.638 0.524444    
factor(month)11 -133.9372    37.2977  -3.591 0.000405 ***
factor(month)12  -33.4865    37.2992  -0.898 0.370271    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 116.4 on 223 degrees of freedom
Multiple R-squared:  0.8793,	Adjusted R-squared:  0.8728 
F-statistic: 135.3 on 12 and 223 DF,  p-value: < 2.2e-16
```

A data.frame: 2 × 3

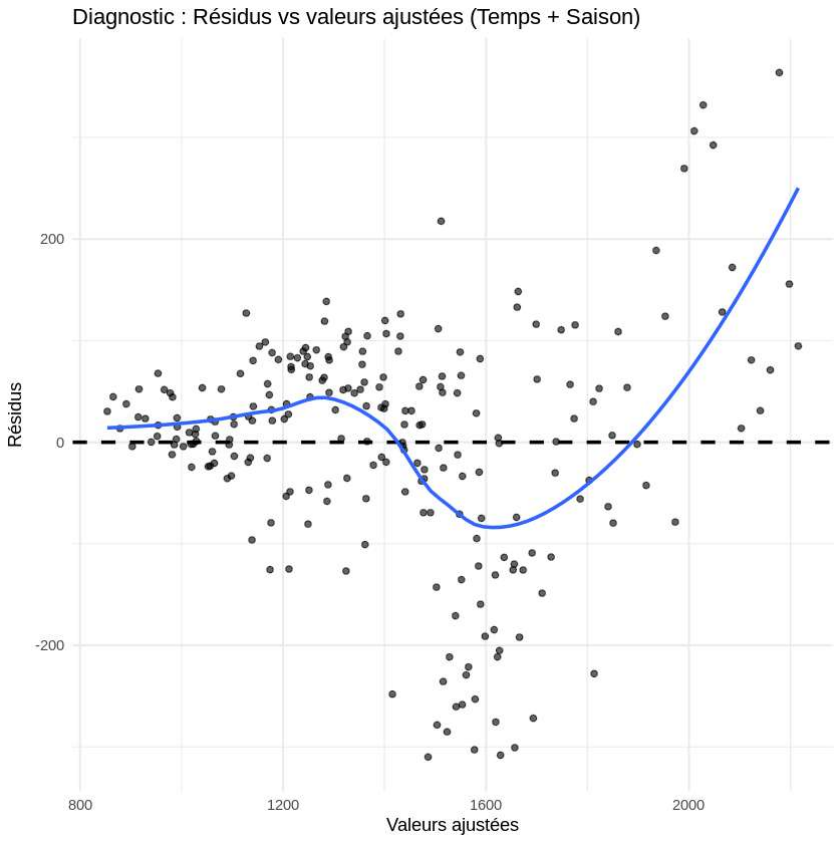
modele	R2	R2_ajuste
<chr>	<dbl>	<dbl>
Temps	0.4431654	0.4407858
Temps + Saison	0.8792726	0.8727761

11) Diagnostic du modèle retenu (m2)

- Diagnostic : on vérifie que les résidus sont globalement centrés autour de 0 et qu'il n'y a pas de structure trop marquée. Une légère non-linéarité peut indiquer des effets non capturés, mais le modèle reste pertinent pour une analyse exploratoire.

```r
diag_df <- data.frame(
  fitted = fitted(m2),
  resid  = residuals(m2)
)

# Résidus vs ajustés
ggplot(diag_df, aes(x = fitted, y = resid)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, linetype = "dashed", linewidth = 1) +
  geom_smooth(method = "loess", se = FALSE) +
  labs(
    title = "Diagnostic : Résidus vs valeurs ajustées (Temps + Saison)",
    x = "Valeurs ajustées",
    y = "Résidus"
  ) +
  theme_minimal()

# QQ-plot
qqnorm(residuals(m2), main = "QQ-plot des résidus (Temps + Saison)")
qqline(residuals(m2))

# Normalité des résidus (indicatif)
shapiro.test(residuals(m2))

```

```
`geom_smooth()` using formula = 'y ~ x'
```




```
	Shapiro-Wilk normality test

data:  residuals(m2)
```