

TD4 ANALYSE DE LA NON-STATIONARITE ET RACINES UNITAIRES

CROCHET Florian et MARTERET Achille

Sommaire

Importation des libraries	3
Chocs permanents vs. chocs transitoires : une analyse exploratoire	4
TS vs. DS : simulation de processus	6
Loi normale centrée de variance 1/4	6
Mise à jour des paramètres	6
Simulation des processus	6
Représentation graphique	6
Loi normale centrée de variance 1/2	9
Mise à jour des paramètres	9
Simulation des processus	9
Représentation graphique	9
Loi normale centrée réduite	11
Mise à jour des paramètres	11
Simulation des processus	11
Représentation graphique	11
Régressions fallacieuses	13
Génération de 5000 séries pour 200 observations chacune	13
Génération de 5000 séries pour 400 observations chacune	14
Génération de 8000 séries pour 200 observations chacune	15
Conclusion	16
Distribution de la statistique de test de Dickey-Fuller pour le modèle sans constante ni tendance via la méthode de Monte Carlo	17
Processus à racine unitaire	17
Processus avec constante $\delta_0 = 5$	20

Processus avec constante $\delta_0 = 10$	23
Analyse de la série temporelle du PIB des USA sur la période 1990-2023	27
Visualisation de la série	27
1. Période de 1990 à 2019	29
1. Mettre la série en time série et restreindre à la période 1990-2019	29
2. Zones ombrées pour les périodes de récession officielle aux US.	30
3. ACF et PACF de la série	31
4. Procédure de test de racine unitaire avec le test de Dickey-Fuller simple puis le test de Dickey-Fuller augmentée.	32
5. Appliquer le test de stationaritée de KPSS.	39
2. Période de 1990 à 2023	40
1. Mettre la série en time série et restreindre à la période 1990-2023	40
2. Zones ombrées pour les périodes de récession officielle aux US (1990-2023)	40
3. ACF et PACF de la série (avec la période 1990-2023)	42
4. Procédure de test de racine unitaire avec le test de Dickey-Fuller simple puis le test de Dickey-Fuller augmenté	43
5. Appliquer le test de stationnarité de KPSS (pour la période 1990-2023)	49
3. Conclusion	50

Importation des libraries

```
library(tidyverse)
library(openxlsx)
library(urca)
library(forecast)
library(tseries)
```

Chocs permanents vs. chocs transitoires : une analyse exploratoire

```
simulate_ar1 <- function(phi, choc_value, choc_time = 100, n = 200) {
  Y <- numeric(n)
  eps <- rnorm(n, mean = 0, sd = 1)

  for (t in 2:n) {
    Y[t] <- phi * Y[t - 1] + eps[t]
    if (t == choc_time) {
      Y[t] <- Y[t] + choc_value
    }
  }
  return(Y)
}

set.seed(132)

phi_vals <- c(0.5, 0.9, 1.0)
chocs <- c(20, 40, -20, -40)

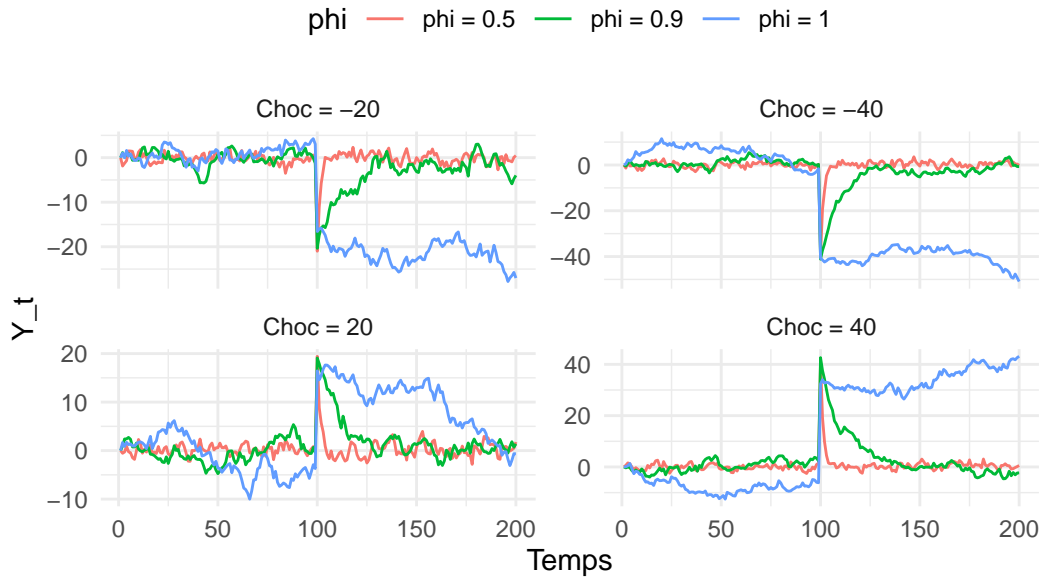
# Stocker les résultats dans une data.frame
results <- data.frame()

for (choc in chocs) {
  for (phi in phi_vals) {
    serie <- simulate_ar1(phi, choc)
    df <- data.frame(
      t = 1:200,
      Y = serie,
      phi = paste0("phi = ", phi),
      choc = paste0("Choc = ", choc)
    )
    results <- rbind(results, df)
  }
}

ggplot(results, aes(x = t, y = Y, color = phi)) +
  geom_line() +
  facet_wrap(~choc, scales = "free_y") +
  labs(title = "Impact de chocs à t=100 pour différentes valeurs de ",
       x = "Temps", y = "Y_t") +
  theme_minimal() +
```

```
theme(legend.position = "top")
```

Impact de chocs à $t=100$ pour différentes valeurs de ϕ



```
simulate_ar1 <- function(phi, choc_value, choc_time = 100, n = 200) {
  Y <- numeric(n)
  eps <- rnorm(n, mean = 0, sd = 1)

  for (t in 2:n) {
    Y[t] <- phi * Y[t - 1] + eps[t]
    if (t == choc_time) {
      Y[t] <- Y[t] + choc_value
    }
  }
  return(Y)
}
```

Les processus avec $\phi = 1$ ne retombe pas sur la tendance ce qui confirme la marche aléatoire.

Pour $\phi = 0.5$, le choc disparaît rapidement. Pour $\phi = 0.9$, il persiste plus longtemps. Pour $\phi = 1.0$, il devient permanent.

Dans le cadre d'une marche aléatoire le choc à un impacte permanent et aléatoire sur les différents processus, tandis que pour les $\phi < 1$, les processus reviennent à leur niveau de tendance initial.

TS vs. DS : simulation de processus

```
set.seed(246)

T <- 200 # Echantillon de 1000 observations

# Processus stochastique
Y_TS <- numeric(T)
Y_DS <- numeric(T)
```

Loi normale centrée de variance 1/4

Mise à jour des paramètres

```
sigma1 <- 1/4

epsilon1 <- rnorm(T, mean = 0, sd = sqrt(sigma1)) # ~ N(0,1/4)
```

Simulation des processus

```
# Premier processus

for (t in 1:T) {
  Y_TS[t] <- 0.2 * t + epsilon1[t]
}

# Second processus

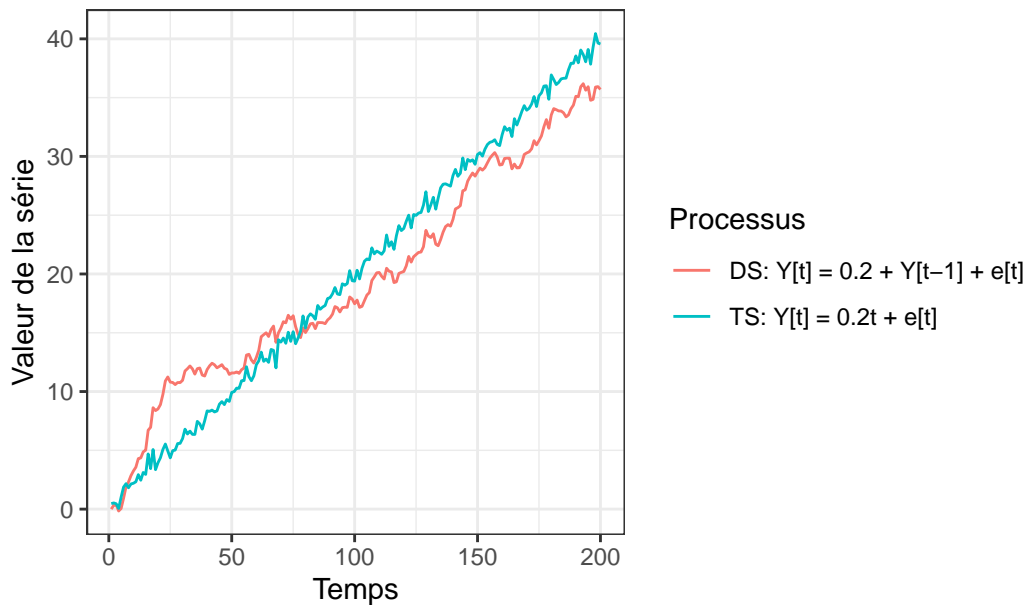
for (t in 2:T) {
  Y_DS[t] <- 0.2 + Y_DS[t - 1] + epsilon1[t]
}
```

Représentation graphique

```
# Base de données
data <- tibble(
  Temps = rep(seq_len(T), times = 2),
  Valeur = c(Y_TS, Y_DS),
  Processus = rep(c("TS: Y[t] = 0.2t + [t]", "DS: Y[t] = 0.2 + Y[t-1] + [t]"), each = T)
)

# Graphique
data |>
  ggplot() +
  aes(x = Temps, y = Valeur, color = Processus) +
  geom_line() +
  labs(title = "Comparaison entre un processus déterministe et un processus stochastique",
       x = "Temps",
       y = "Valeur de la série") +
  theme_bw()
```

Comparaison entre un processus déterministe et un processus



La série TS est stationnaire autour d'une tendance déterministe (une tendance linéaire de pente $10/50=0.2$). Les chocs ε_t n'ont pas d'effet permanent sur la série : après un choc, la série revient à sa tendance.

La série DS contient une racine unitaire. Un choc ε_t a un effet permanent : la série ne revient pas à une tendance fixe après un choc.

Loi normale centrée de variance 1/2

Mise à jour des paramètres

```
sigma2 <- 1/2  
  
epsilon2 <- rnorm(T, mean = 0, sd = sqrt(sigma2)) # ~ N(0,1/2)
```

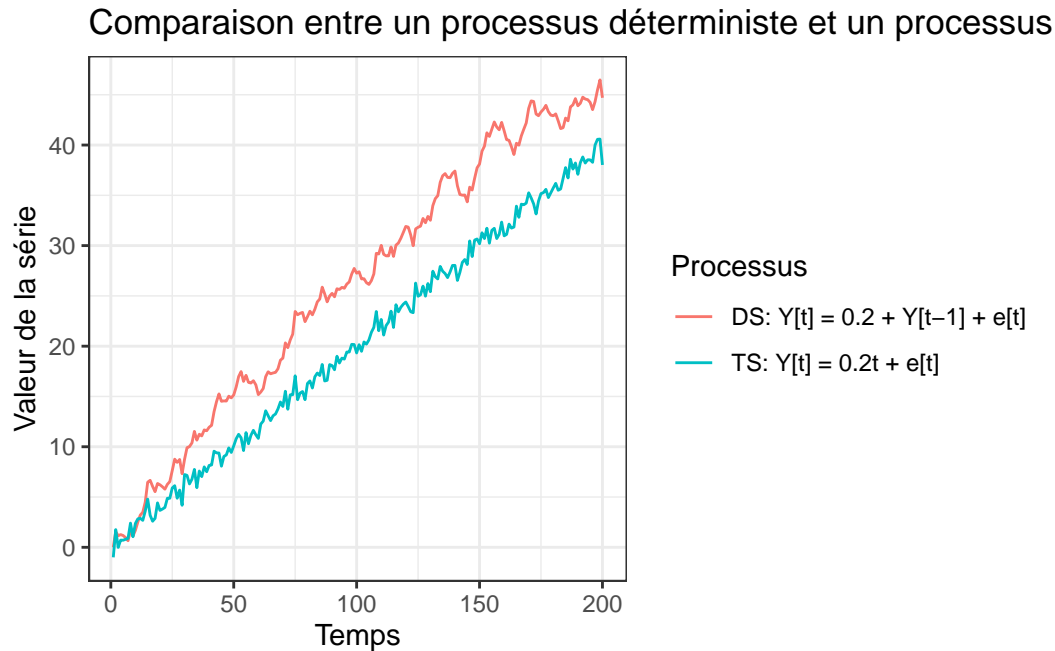
Simulation des processus

```
# Premier processus  
  
for (t in 1:T) {  
  Y_TS[t] <- 0.2 * t + epsilon2[t]  
}  
  
# Second processus  
  
for (t in 2:T) {  
  Y_DS[t] <- 0.2 + Y_DS[t - 1] + epsilon2[t]  
}
```

Représentation graphique

```
# Base de données  
data <- tibble(  
  Temps = rep(seq_len(T), times = 2),  
  Valeur = c(Y_TS, Y_DS),  
  Processus = rep(c("TS: Y[t] = 0.2t + [t]", "DS: Y[t] = 0.2 + Y[t-1] + [t]"), each = T)  
)  
  
# Graphique  
data |>  
  ggplot() +  
  aes(x = Temps, y = Valeur, color = Processus) +  
  geom_line() +
```

```
labs(title = "Comparaison entre un processus déterministe et un processus stochastique",
     x = "Temps",
     y = "Valeur de la série") +
theme_bw()
```



La série TS est stationnaire autour d'une tendance déterministe (une tendance linéaire de pente $10/50=0.2$). Les chocs ε_t n'ont pas d'effet permanent sur la série : après un choc, la série revient à sa tendance.

La série DS contient une racine unitaire. Un choc ε_t a un effet permanent : la série ne revient pas à une tendance fixe après un choc.

En augmentant la variance des résidus de $1/4$ à $1/2$, nous avons accru l'impact des chocs. Nous observons ainsi que les variations des deux séries sont plus importantes.

Loi normale centrée réduite

Mise à jour des paramètres

```
sigma3 <- 1  
epsilon3 <- rnorm(T, mean = 0, sd = sqrt(sigma3)) # ~ N(0,1)
```

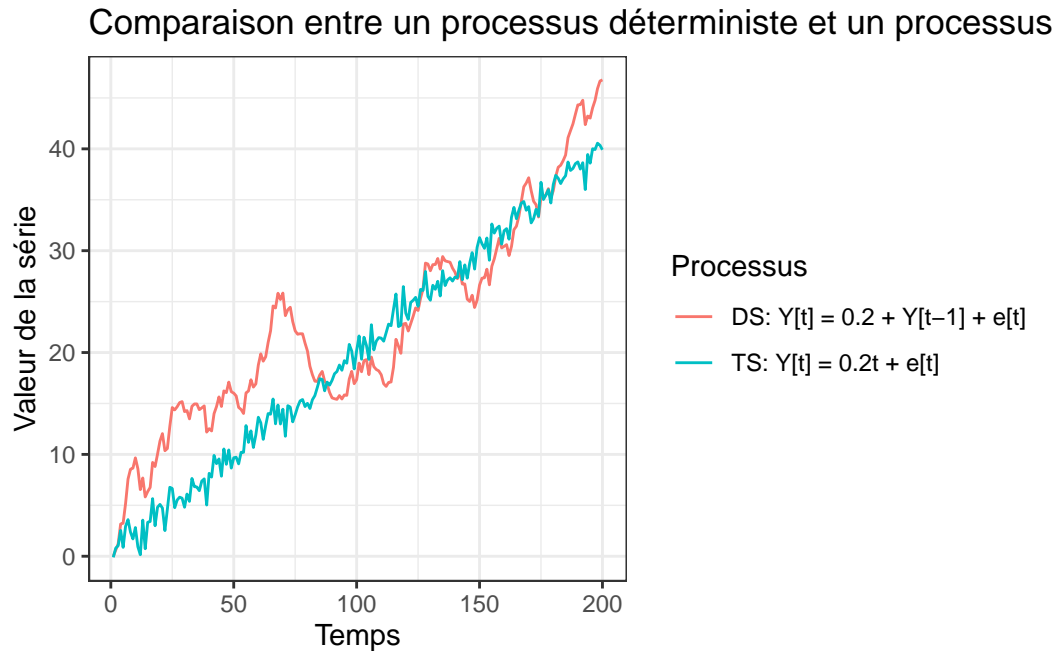
Simulation des processus

```
# Premier processus  
for (t in 1:T) {  
  Y_TS[t] <- 0.2 * t + epsilon3[t]  
}  
  
# Second processus  
for (t in 2:T) {  
  Y_DS[t] <- 0.2 + Y_DS[t - 1] + epsilon3[t]  
}
```

Représentation graphique

```
# Base de données  
data <- tibble(  
  Temps = rep(seq_len(T), times = 2),  
  Valeur = c(Y_TS, Y_DS),  
  Processus = rep(c("TS: Y[t] = 0.2t + [t]", "DS: Y[t] = 0.2 + Y[t-1] + [t]"), each = T)  
)  
  
# Graphique  
data |>  
  ggplot() +  
  aes(x = Temps, y = Valeur, color = Processus) +  
  geom_line() +
```

```
labs(title = "Comparaison entre un processus déterministe et un processus stochastique",
     x = "Temps",
     y = "Valeur de la série") +
theme_bw()
```



La série TS est stationnaire autour d'une tendance déterministe (une tendance linéaire de pente $10/50=0.2$). Les chocs ε_t n'ont pas d'effet permanent sur la série : après un choc, la série revient à sa tendance.

La série DS contient une racine unitaire. Un choc ε_t a un effet permanent : la série ne revient pas à une tendance fixe après un choc.

En augmentant la variance des résidus $1/2$ à 1 , nous avons accru l'impact des chocs. Nous observons ainsi que les variations des deux séries sont plus importantes.

Régressions fallacieuses

Génération de 5000 séries pour 200 observations chacune

```
set.seed(123)

# Paramètres
n_series <- 5000
n_obs <- 200
alpha <- 0.05

# Fonction pour générer une marche aléatoire
marche_aleatoire <- function(n) {
  cumsum(rnorm(n))
}

# Stockage des p-values
p_values <- numeric(n_series)

# Boucle pour générer les séries et effectuer les régressions
for (i in 1:n_series) {
  Xt <- marche_aleatoire(n_obs)
  Yt <- marche_aleatoire(n_obs)

  model <- lm(Yt ~ Xt)
  p_values[i] <- summary(model)$coefficients[2, 4] # p-value associée à 1
}

# Pourcentage de rejets de H0
rejection_rate <- mean(p_values < alpha) * 100

cat("Pourcentage de rejets de H0 au seuil de 5%:", rejection_rate, "%\n")
```

Pourcentage de rejets de H0 au seuil de 5%: 83.52 %

Nos deux processus $\{X_t\}$ et $\{Y_t\}$ générés artificiellement comme deux marches aléatoires indépendantes, devrait être indépendantes, il ne devrait pas y avoir de corrélation entre les deux séries. Ainsi, si on estime $Y_t = 0 + 1X_t + \epsilon_t$, 1 devrait être différent de 0 que dans 5% des cas. Or, on remarque dans nos séries que l'hypothèse $H_0 : 1 = 0$ au taux d'erreur de 5%, obtient un pourcentage de rejet de 83.52% pour 200 observations et 5000 séries testés.

Génération de 5000 séries pour 400 observations chacune

```
set.seed(123)

# Paramètres
n_series <- 5000
n_obs <- 400      # Nous passons de 200 à 400 observations.
alpha <- 0.05

# Fonction pour générer une marche aléatoire
marche_aleatoire <- function(n) {
  cumsum(rnorm(n))
}

# Stockage des p-values
p_values <- numeric(n_series)

# Boucle pour générer les séries et effectuer les régressions
for (i in 1:n_series) {
  Xt <- marche_aleatoire(n_obs)
  Yt <- marche_aleatoire(n_obs)

  model <- lm(Yt ~ Xt)
  p_values[i] <- summary(model)$coefficients[2, 4] # p-value associée à 1
}

# Pourcentage de rejets de H0
rejection_rate <- mean(p_values < alpha) * 100

cat("Pourcentage de rejets de H0 au seuil de 5%:", rejection_rate, "%\n")
```

Pourcentage de rejets de H0 au seuil de 5%: 88.16 %

Lorsqu'on augmente le nombre d'observations le pourcentage de rejet augmente. Pour 400 observations le pourcentage de rejet passe de 83.52% à 88.16%

Génération de 8000 séries pour 200 observations chacune

```
set.seed(123)

# Paramètres
n_series <- 8000 # Nous passons de 5000 à 8000 observations.
n_obs <- 200
alpha <- 0.05

# Fonction pour générer une marche aléatoire
marche_aleatoire <- function(n) {
  cumsum(rnorm(n))
}

# Stockage des p-values
p_values <- numeric(n_series)

# Boucle pour générer les séries et effectuer les régressions
for (i in 1:n_series) {
  Xt <- marche_aleatoire(n_obs)
  Yt <- marche_aleatoire(n_obs)

  model <- lm(Yt ~ Xt)
  p_values[i] <- summary(model)$coefficients[2, 4] # p-value associée à 1
}

# Pourcentage de rejets de H0
rejection_rate <- mean(p_values < alpha) * 100

cat("Pourcentage de rejets de H0 au seuil de 5%:", rejection_rate, "%\n")
```

Pourcentage de rejets de H0 au seuil de 5%: 82.9875 %

Lorsqu'on augmente le nombre de séries générées, passant de 5000 à 8000, le pourcentage de rejets au seuil de 5% diminue légèrement, passant à 82.9875%.

Les résultats dépendent donc plus fortement du nombre d'observation générée, que du nombre de série générée.

Conclusion

Les résultats montrent qu'en présence de marches aléatoires, la régression produit un taux de rejet de H_0 bien supérieur aux 5% attendus sous l'hypothèse nulle, ce qui illustre une corrélation fallacieuse.

Lorsque le nombre d'observations par série augmente, le taux de rejet s'accroît (de 83,52% à 88,16% pour 200 à 400 observations), ce qui suggère que la dépendance temporelle accrue amplifie la fausse corrélation.

En revanche, l'augmentation du nombre de séries simulées ne modifie pas significativement le phénomène (82,99% pour 8000 séries contre 83,52% pour 5000), confirmant que le biais provient de la structure des données et non du nombre de simulations.

Distribution de la statistique de test de Dickey-Fuller pour le modèle sans constante ni tendance via la méthode de Monte Carlo

Processus à racine unitaire

```
set.seed(246)

# Paramètres
nb_obs <- 100
nb_sim <- 10000

# Stockage des t-statistiques
t_stats <- numeric(nb_sim)

# Simulation du processus racine unitaire
for (i in 1:nb_sim) {

  epsilon <- rnorm(nb_obs, mean = 0, sd = 1) # Résidus normaux centrés réduits
  Y <- numeric(nb_obs)

  Y[1] <- epsilon[1] # Première observation

  for (t in 2:nb_obs) {
    Y[t] <- Y[t-1] + epsilon[t] # Processus racine unitaire
  }

  # Régression de Yt sur Yt-1 pour estimer le paramètre phi_1 et son écart-type
  modele <- lm(Y[2:nb_obs] ~ Y[1:(nb_obs-1)] - 1)

  phi_1 <- coef(modele)[1] # Coefficient estimé phi_1
  gamma <- phi_1 - 1 # Calcul de gamma estimé

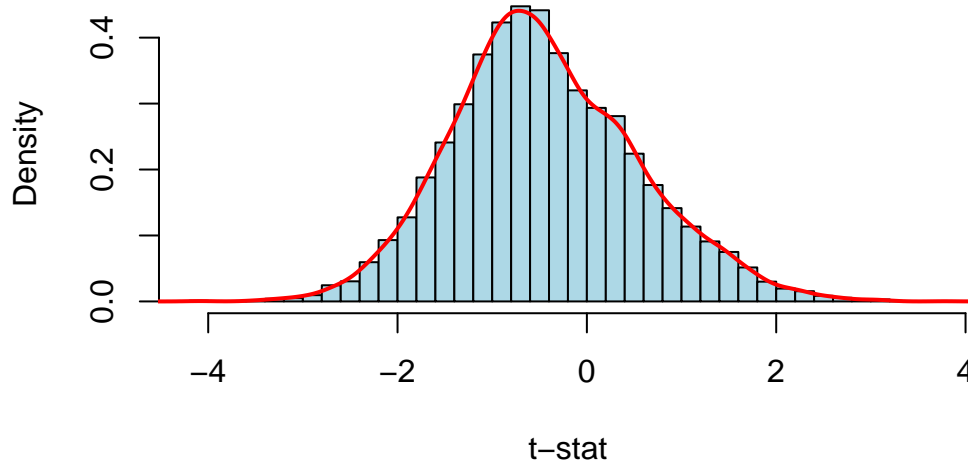
  # Ecart-type de gamma (égal à l'écart-type de phi_1)
  se_gamma <- summary(modele)$coefficients[1, 2]

  # Statistique de test
  t_stats[i] <- gamma / se_gamma
}

# Estimation de la densité par noyau pour visualiser la distribution
densite <- density(t_stats)
```

```
# Histogramme avec la courbe de densité lissée
hist(
  t_stats,
  breaks = 50,
  probability = TRUE,
  col = "lightblue",
  main = "Distribution des t-statistiques avec lissage par noyau",
  xlab = "t-stat"
)
lines(
  densite,
  col = "red",
  lwd = 2
)
```

Distribution des t-statistiques avec lissage par noyau



```
# Calcul des valeurs critiques pour 10%, 5% et 1%
val_crit_est_10 <- quantile(t_stats, 0.10)
val_crit_est_5  <- quantile(t_stats, 0.05)
val_crit_est_1  <- quantile(t_stats, 0.01)

cat("Valeur critique unilatérale à 10% :", val_crit_est_10, "\n")
```

Valeur critique unilatérale à 10% : -1.637605

```
cat("Valeur critique unilatérale à 5% :", val_crit_est_5, "\n")
```

Valeur critique unilatérale à 5% : -1.957615

```
cat("Valeur critique unilatérale à 1% :", val_crit_est_1, "\n")
```

Valeur critique unilatérale à 1% : -2.572868

```
# Valeurs critiques réelles pour n = 100 et p = 10%, p = 5% et p = 1%
critical_values <- unitrootTable(trend = "nc", statistic = "t")

val_crit_10 <- critical_values["100", "0.100"]
val_crit_5  <- critical_values["100", "0.050"]
val_crit_1  <- critical_values["100", "0.010"]

# Tableau comparatif
tableau_comparatif <- tibble(
  "Seuil" = c("10%", "5%", "1%"),
  "Valeur critique estimée" = c(val_crit_est_10, val_crit_est_5, val_crit_est_1),
  "Valeur critique réelle" = c(val_crit_10, val_crit_5, val_crit_1)
)

tableau_comparatif
```

```
# A tibble: 3 x 3
  Seuil `Valeur critique estimée` `Valeur critique réelle`
  <chr>          <dbl>          <dbl>
1 10%          -1.64          -1.62
2 5%           -1.96          -1.94
3 1%           -2.57          -2.59
```

Grâce à la méthode de Monte Carlo, les valeurs obtenues par simulation d'un processus à racine unitaire, sans constante ni tendance, sont quasi identiques à celles figurant dans la table de Dickey-Fuller. Plus le nombre de simulations est élevé, plus les statistiques obtenues convergent vers celles présentes dans la table.

Processus avec constante $\delta_0 = 5$

```
set.seed(246)

# Paramètres
nb_obs <- 100
nb_sim <- 10000

# Stockage des t-statistiques
t_stats <- numeric(nb_sim)

# Simulation du processus avec constante ( $\delta_0 = 5$ )
delta_0 <- 5
for (i in 1:nb_sim) {

  epsilon <- rnorm(nb_obs, mean = 0, sd = 1) # Résidus normaux centrés réduits
  Y <- numeric(nb_obs)

  Y[1] <- delta_0 + epsilon[1] # Première observation avec constante

  for (t in 2:nb_obs) {
    Y[t] <- delta_0 + Y[t-1] + epsilon[t] # Processus avec constante
  }

  # Création de la variable de tendance
  trend <- 1:nb_obs # Variable temporelle croissante

  # pour tester un modèle avec constante, il est préférable d'adopter une spécification
  # avec constante et tendance
  modele <- lm(Y[2:nb_obs] ~ Y[1:(nb_obs-1)] + trend[2:nb_obs])

  phi_1 <- coef(modele)[2] # Coefficient estimé  $\phi_1$  (sur  $Y_{t-1}$ )
  gamma <- phi_1 - 1 # Calcul de  $\gamma$  estimé

  # Ecart-type de  $\gamma$  (égal à l'écart-type de  $\phi_1$ )
  se_gamma <- summary(modele)$coefficients[2, 2]

  # Statistique de test
  t_stats[i] <- gamma / se_gamma
}

# Estimation de la densité par noyau pour visualiser la distribution
```

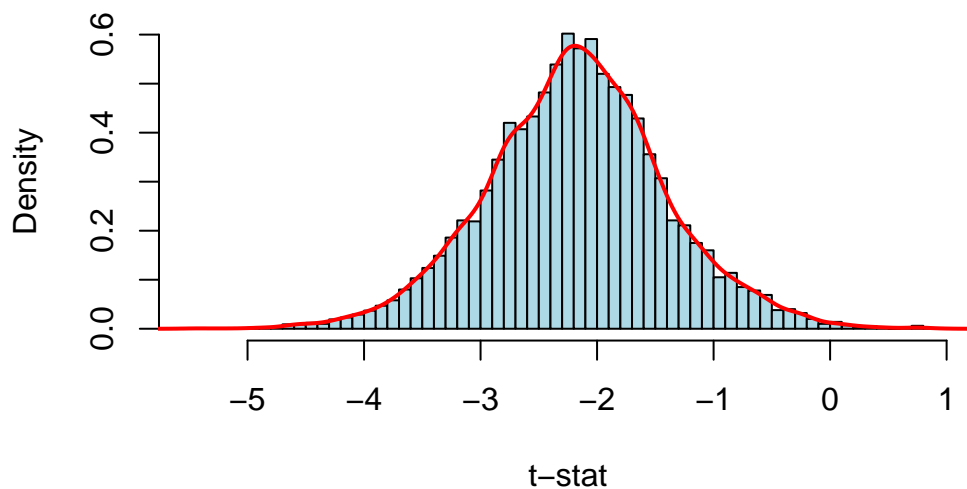
```

densite <- density(t_stats)

# Histogramme avec la courbe de densité lissée
hist(
  t_stats,
  breaks = 50,
  probability = TRUE,
  col = "lightblue",
  main = "Distribution des t-statistiques avec lissage par noyau",
  xlab = "t-stat"
)
lines(
  densite,
  col = "red",
  lwd = 2
)

```

Distribution des t-statistiques avec lissage par noyau



```

# Calcul des valeurs critiques pour 10%, 5% et 1%
val_crit_est_10 <- quantile(t_stats, 0.10)
val_crit_est_5  <- quantile(t_stats, 0.05)
val_crit_est_1  <- quantile(t_stats, 0.01)

```

```
cat("Valeur critique unilatérale à 10% :", val_crit_est_10, "\n")
```

Valeur critique unilatérale à 10% : -3.156088

```
cat("Valeur critique unilatérale à 5% :", val_crit_est_5, "\n")
```

Valeur critique unilatérale à 5% : -3.451863

```
cat("Valeur critique unilatérale à 1% :", val_crit_est_1, "\n")
```

Valeur critique unilatérale à 1% : -4.074694

```
# Valeurs critiques réelles pour n = 100 et p = 10%, p = 5% et p = 1%
critical_values <- unitrootTable(trend = "ct", statistic = "t")

val_crit_10 <- critical_values["100", "0.100"]
val_crit_5  <- critical_values["100", "0.050"]
val_crit_1  <- critical_values["100", "0.010"]

# Tableau comparatif
tableau_comparatif <- tibble(
  "Seuil" = c("10%", "5%", "1%"),
  "Valeur critique estimée" = c(val_crit_est_10, val_crit_est_5, val_crit_est_1),
  "Valeur critique réelle" = c(val_crit_10, val_crit_5, val_crit_1)
)

tableau_comparatif
```

```
# A tibble: 3 x 3
  Seuil `Valeur critique estimée` `Valeur critique réelle`
  <chr>          <dbl>          <dbl>
1 10%           -3.16           -3.15
2 5%            -3.45           -3.46
3 1%            -4.07           -4.05
```

Nous remarquons que les valeurs critiques estimées sont proches de celles de la table de Dickey-Fuller.

Processus avec constante $\delta_0 = 10$

```
set.seed(246)

# Paramètres
nb_obs <- 100
nb_sim <- 10000

# Stockage des t-statistiques
t_stats <- numeric(nb_sim)

# Simulation du processus avec constante ( $\delta_0 = 10$ )
delta_0 <- 10
for (i in 1:nb_sim) {

  epsilon <- rnorm(nb_obs, mean = 0, sd = 1) # Résidus normaux centrés réduits
  Y <- numeric(nb_obs)

  Y[1] <- delta_0 + epsilon[1] # Première observation avec constante

  for (t in 2:nb_obs) {
    Y[t] <- delta_0 + Y[t-1] + epsilon[t] # Processus avec constante
  }

  # Création de la variable de tendance
  trend <- 1:nb_obs # Variable temporelle croissante

  # pour tester un modèle avec constante, il est préférable d'adopter une spécification
  # avec constante et tendance
  modele <- lm(Y[2:nb_obs] ~ Y[1:(nb_obs-1)] + trend[2:nb_obs])

  phi_1 <- coef(modele)[2] # Coefficient estimé  $\phi_1$  (sur  $Y_{t-1}$ )
  gamma <- phi_1 - 1 # Calcul de  $\gamma$  estimé

  # Ecart-type de  $\gamma$  (égal à l'écart-type de  $\phi_1$ )
  se_gamma <- summary(modele)$coefficients[2, 2]

  # Statistique de test
  t_stats[i] <- gamma / se_gamma
}

# Estimation de la densité par noyau pour visualiser la distribution
```

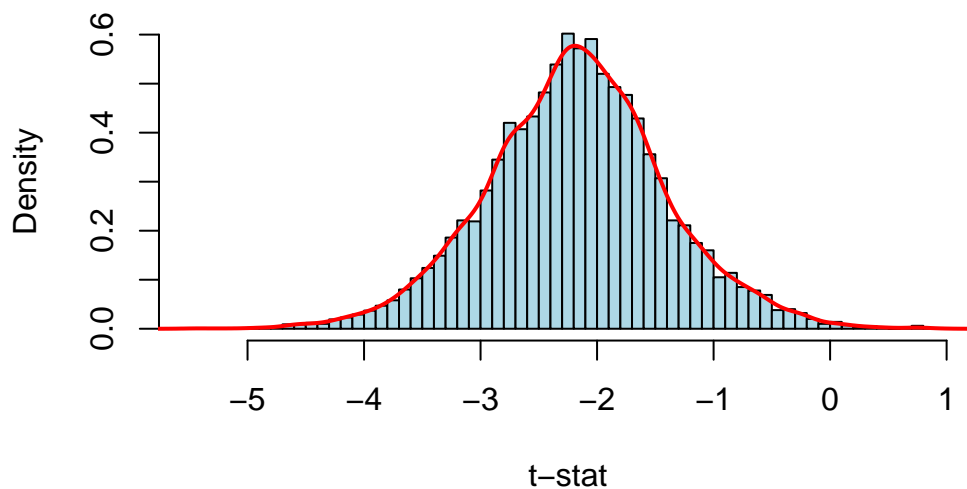
```

densite <- density(t_stats)

# Histogramme avec la courbe de densité lissée
hist(
  t_stats,
  breaks = 50,
  probability = TRUE,
  col = "lightblue",
  main = "Distribution des t-statistiques avec lissage par noyau",
  xlab = "t-stat"
)
lines(
  densite,
  col = "red",
  lwd = 2
)

```

Distribution des t-statistiques avec lissage par noyau



```

# Calcul des valeurs critiques pour 10%, 5% et 1%
val_crit_est_10 <- quantile(t_stats, 0.10)
val_crit_est_5  <- quantile(t_stats, 0.05)
val_crit_est_1  <- quantile(t_stats, 0.01)

```



```
cat("Valeur critique unilatérale à 10% :", val_crit_est_10, "\n")
```

Valeur critique unilatérale à 10% : -3.156088

```
cat("Valeur critique unilatérale à 5% :", val_crit_est_5, "\n")
```

Valeur critique unilatérale à 5% : -3.451863

```
cat("Valeur critique unilatérale à 1% :", val_crit_est_1, "\n")
```

Valeur critique unilatérale à 1% : -4.074694

```
# Valeurs critiques réelles pour n = 100 et p = 10%, p = 5% et p = 1%
critical_values <- unitrootTable(trend = "ct", statistic = "t")

val_crit_10 <- critical_values["100", "0.100"]
val_crit_5  <- critical_values["100", "0.050"]
val_crit_1  <- critical_values["100", "0.010"]

# Tableau comparatif
tableau_comparatif <- tibble(
  "Seuil" = c("10%", "5%", "1%"),
  "Valeur critique estimée" = c(val_crit_est_10, val_crit_est_5, val_crit_est_1),
  "Valeur critique réelle" = c(val_crit_10, val_crit_5, val_crit_1)
)

tableau_comparatif
```

```
# A tibble: 3 x 3
  Seuil `Valeur critique estimée` `Valeur critique réelle`
  <chr>          <dbl>          <dbl>
1 10%           -3.16           -3.15
2 5%            -3.45           -3.46
3 1%            -4.07           -4.05
```

Nous remarquons que les valeurs critiques estimées sont proches de celles de la table de Dickey-Fuller.

La valeur de la constante dans les simulations n'a pas d'impact sur les résultats du test. En comparant les valeurs critiques estimées pour $\delta_0 = 5$ et $\delta_0 = 10$, on observe qu'elles restent

identiques et proches des valeurs critiques théoriques issues de la table de Dickey-Fuller. Cela confirme que la constante influence le niveau moyen du processus Y_t , mais n'affecte ni la statistique de test ni la significativité des résultats.

Analyse de la série temporelle du PIB des USA sur la période 1990-2023

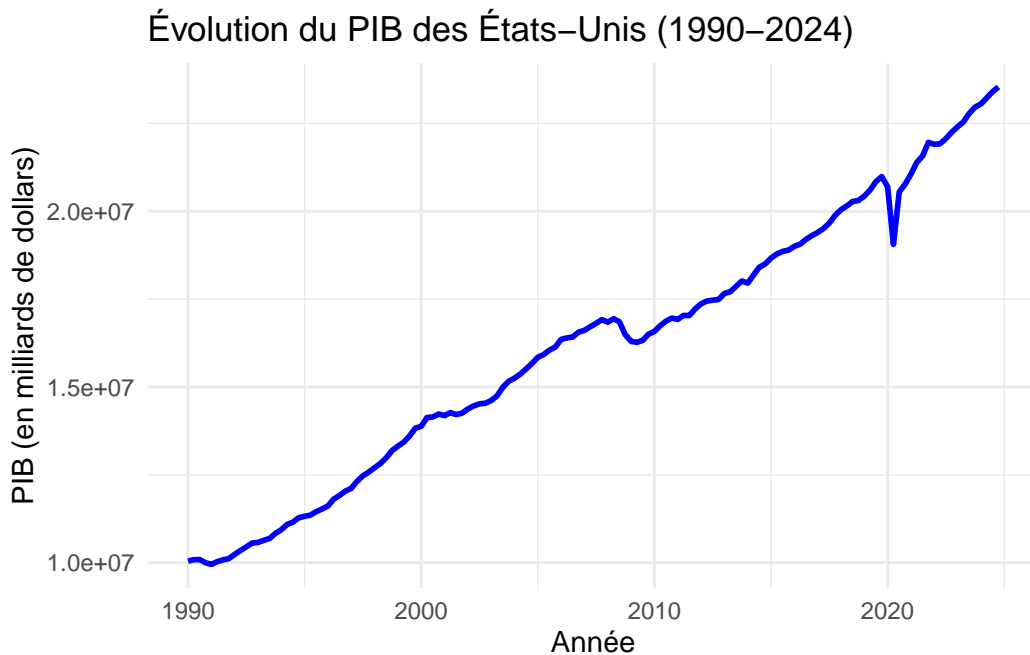
Visualisation de la série

```
Classes 'data.table' and 'data.frame': 140 obs. of 2 variables:
 $ Date: Date, format: "1990-01-01" "1990-04-01" ...
 $ PIB : num 10047386 10083855 10090569 9998704 9951916 ...
- attr(*, ".internal.selfref")=<externalptr>
```

Il est important de vérifier que la Date est bien en format 'Date' !

```
ggplot(df_Y_us, aes(x=Date, y=PIB)) +
  geom_line(color="blue", size=1) +
  labs(title="Évolution du PIB des États-Unis (1990-2024)",
       x="Année",
       y="PIB (en milliards de dollars)") +
  theme_minimal()
```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.



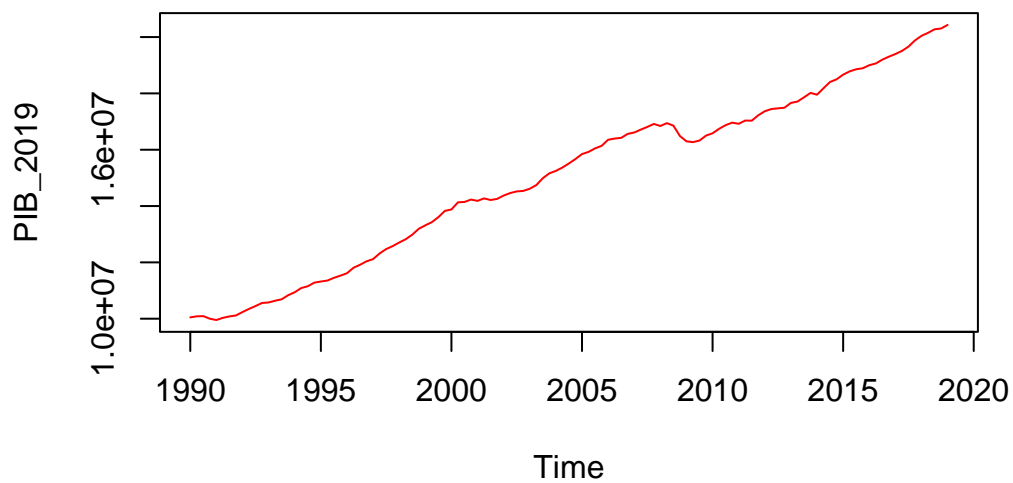
L'évolution du PIB des États-Unis n'a fait que d'augmenter depuis 1990, ce qui est caractéristique d'une tendance à la hausse. On remarque également quelques fortes baisses à certaines périodes. Aussi, dû au fort choc et arrêt de la production en 2019 en raison de la crise COVID, on va restreindre la période à 1990-2019.

1. Période de 1990 à 2019

1. Mettre la série en time série et restreindre à la période 1990-2019

```
PIB_2019 <- ts(data = df_Y_us$PIB, start = c(1990, 1), end = c(2019, 1), frequency = 4)
plot(PIB_2019, main="Time-Series du PIB des USA sur la période 1990 à 2019", col = "red")
```

Time-Series du PIB des USA sur la période 1990 à 2019



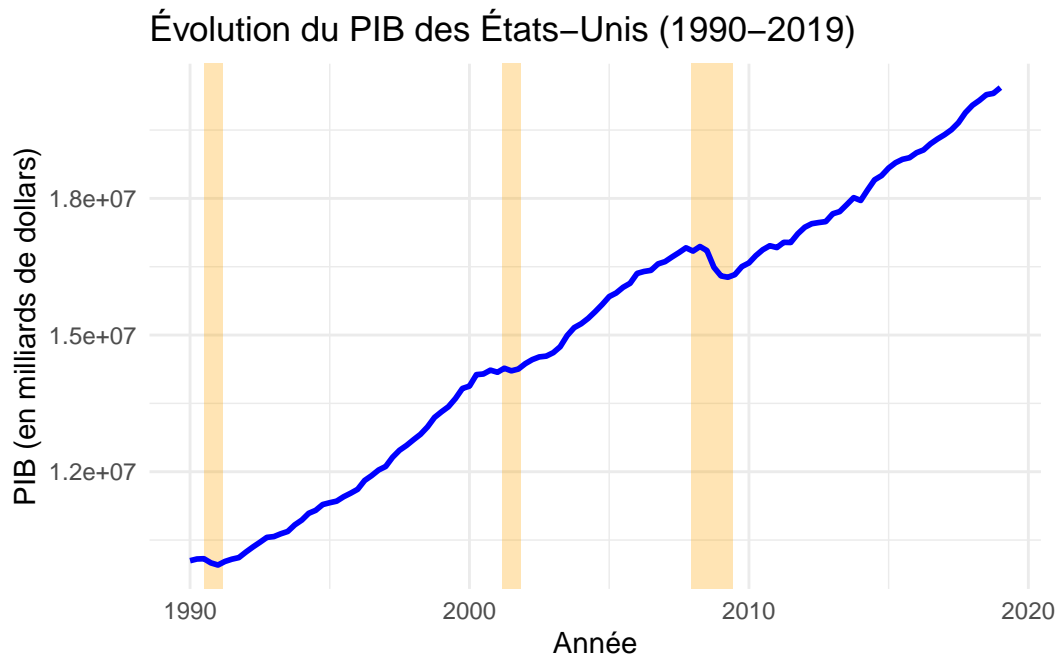
La série temporelle met en évidence Un premier choc baissier visible en 2001, dû aux attentats du 11 septembre 2001. On remarque aussi la chute de la tendance haussière du PIB des USA en 2008, dû à la crise financière et bancaire des subprimes. Ces chocs on eu un impacte significatif sur le PIB au USA.

2. Zones ombrées pour les périodes de récession officielle aux US.

```
# Jusqu'à la période 2019
us_df_filtre <- df_Y_us |>
  filter(Date <= as.Date("2019-01-01"))

# Exemple des périodes de récession officielle aux US (source : NBER)
recessions <- data.frame(
  start = as.Date(c("1990-07-01", "2001-03-01", "2007-12-01")),
  end   = as.Date(c("1991-03-01", "2001-11-01", "2009-06-01"))
)

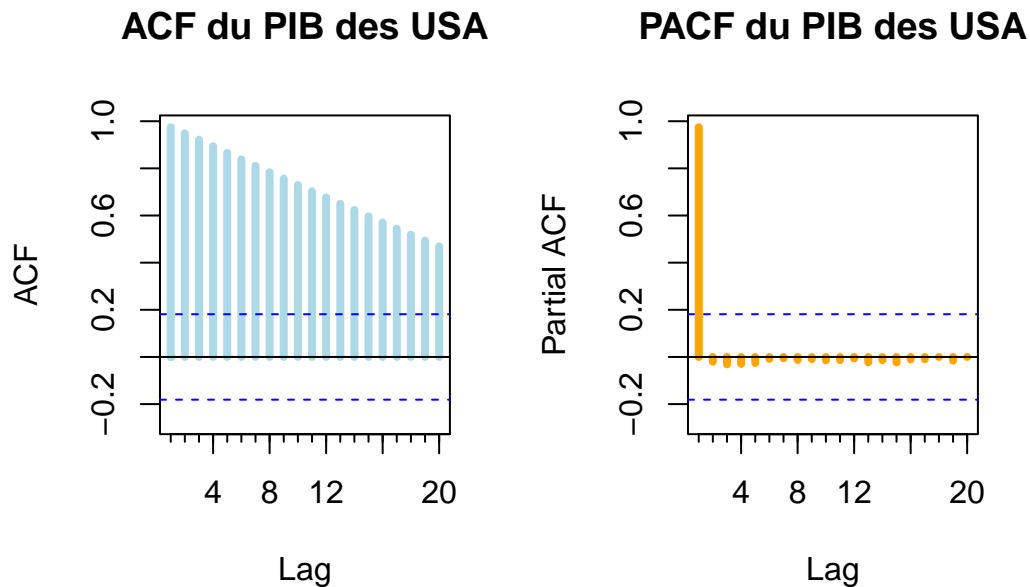
# Tracer le graphique avec zones de récession
ggplot(us_df_filtre, aes(x = Date, y = PIB)) +
  # Ajout des zones de récession
  geom_rect(data = recessions,
            aes(xmin = start, xmax = end, ymin = -Inf, ymax = Inf),
            fill = "orange", alpha = 0.3, inherit.aes = FALSE) +
  # Courbe du PIB
  geom_line(color = "blue", size = 1) +
  # Titres et labels
  labs(title = "Évolution du PIB des États-Unis (1990-2019)",
        x = "Année",
        y = "PIB (en milliards de dollars)") +
  # Thème épuré
  theme_minimal()
```



Les zones ombrées montrent bien les différents points de récessions officielle fourni par la National Bureau of Economic Research | NBER. La récession la plus marquante fut celle de 2008, qui dura le plus longtemps.

3. ACF et PACF de la série

```
par(mfrow=c(1,2))  
Acf(PIB_2019, main = "ACF du PIB des USA", col="lightblue", lwd = 4)  
Pacf(PIB_2019, main = "PACF du PIB des USA", col="orange", lwd = 4)
```



```
par(mfrow=c(1,1))
```

La PACF montre que le PIB des USA dépend fortement du PIB de l'année précédente. L'ACF montre une décroissance lente et progressive des autocorrélations, cette décroissance graduelle sans coupure nette est caractéristique d'un processus non-stationnaire.

La série présente une forte tendance, ce qui est signe de non-stationnarité. Elle présente aussi une structure de dépendance où chaque observation est fortement liée aux observations précédentes et un comportement compatible avec un processus autorégressif d'ordre 1 (AR(1)) ou un modèle ARIMA avec composante intégrée.

4. Procédure de test de racine unitaire avec le test de Dickey-Fuller simple puis le test de Dickey-Fuller augmentée.

On va ainsi chercher à savoir si l'hypothèse nulle H_0 (la série est non stationnaire, elle suit un processus de racine unitaire) est rejetée.

La stratégie séquentielle pour le test de racine unitaire consiste à :

1. Tester un modèle avec tendance et constante
2. Si on ne rejette pas H_0 (présence de racine unitaire) → Tester un modèle avec seulement une constante
3. Si on ne rejette toujours pas H_0 → Tester un modèle sans constante ni tendance

a). Test de Dickey-Fuller Simple (DF) avec approche séquentielle

Modèle avec tendance et constante :

```
# Test de Dickey-Fuller simple
df_trend_const <- ur.df(PIB_2019, type = "trend", lags = 0)
summary(df_trend_const)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression trend

Call:

```
lm(formula = z.diff ~ z.lag.1 + 1 + tt)
```

Residuals:

Min	1Q	Median	3Q	Max
-451397	-38383	11838	45178	178159

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.367e+05	2.090e+05	1.611	0.110
z.lag.1	-2.639e-02	2.136e-02	-1.236	0.219
tt	2.541e+03	1.930e+03	1.317	0.191

Residual standard error: 87300 on 113 degrees of freedom

Multiple R-squared: 0.01784, Adjusted R-squared: 0.0004609

F-statistic: 1.027 on 2 and 113 DF, p-value: 0.3616

Value of test-statistic is: -1.2357 41.3392 1.0265

Critical values for test statistics:

	1pct	5pct	10pct
tau3	-3.99	-3.43	-3.13
phi2	6.22	4.75	4.07
phi3	8.43	6.49	5.47

test-statistic = -1.2357 > tau3 pour 1%, 5%, et 10% donc on ne peut pas rejeter H0, alors on continue avec le modèle suivant.

Modèle avec seulement une constante :

```
df_const <- ur.df(PIB_2019, type = "drift", lags = 0)
summary(df_const)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression drift

Call:

```
lm(formula = z.diff ~ z.lag.1 + 1)
```

Residuals:

Min	1Q	Median	3Q	Max
-461273	-38702	10102	49978	164935

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.683e+04	4.113e+04	1.625	0.107
z.lag.1	1.513e-03	2.688e-03	0.563	0.575

Residual standard error: 87580 on 114 degrees of freedom

Multiple R-squared: 0.00277, Adjusted R-squared: -0.005977

F-statistic: 0.3167 on 1 and 114 DF, p-value: 0.5747

Value of test-statistic is: 0.5628 60.7504

Critical values for test statistics:

	1pct	5pct	10pct
tau2	-3.46	-2.88	-2.57
phi1	6.52	4.63	3.81

test-statistic = 0.5628 > tau2 pour 1%, 5%, et 10% donc on ne peut pas rejeter H0, ce qui signifie que la racine unitaire est toujours présente, alors on continue avec le modèle suivant.

Modèle sans tendance ni constante :

```
df_none <- ur.df(PIB_2019, type = "none", lags = 0)
summary(df_none)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression none

Call:

```
lm(formula = z.diff ~ z.lag.1 - 1)
```

Residuals:

Min	1Q	Median	3Q	Max
-466609	-30427	7700	51809	172343

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
z.lag.1	0.0057946	0.0005353	10.82	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 88210 on 115 degrees of freedom

Multiple R-squared: 0.5047, Adjusted R-squared: 0.5004

F-statistic: 117.2 on 1 and 115 DF, p-value: < 2.2e-16

Value of test-statistic is: 10.8254

Critical values for test statistics:

	1pct	5pct	10pct
tau1	-2.58	-1.95	-1.62

Test-statistic > tau1, alors H0 est acceptée ici, la série est non stationnaire, il faudra donc différencier la série.

b). Test de Dickey-Fuller Augmenté (ADF) avec approche séquentielle

Lorsque l'autocorrélation des résidus est importante, on utilise l'ADF avec sélection automatique du lag optimal (généralement par le critère AIC ou BIC).

Modèle avec tendance et constante :

```
# Test de Dickey-Fuller augmenté
adf_trend_const <- ur.df(PIB_2019, type = "trend", selectlags = "AIC")
summary(adf_trend_const)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression trend

Call:

```
lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-378591	-45360	2065	52694	195588

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.987e+05	1.952e+05	2.042	0.0435 *
z.lag.1	-3.574e-02	2.004e-02	-1.784	0.0772 .
tt	3.300e+03	1.811e+03	1.823	0.0710 .
z.diff.lag	3.834e-01	8.758e-02	4.378	2.72e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 81270 on 111 degrees of freedom

Multiple R-squared: 0.1612, Adjusted R-squared: 0.1385

F-statistic: 7.11 on 3 and 111 DF, p-value: 0.0002057

Value of test-statistic is: -1.7839 10.3847 1.6797

Critical values for test statistics:

	1pct	5pct	10pct
tau3	-3.99	-3.43	-3.13

```
phi2  6.22  4.75  4.07
phi3  8.43  6.49  5.47
```

test-statistic : $-1.7839 > \tau_3$, donc H_0 n'est pas rejeté, alors on continue avec le modèle suivant.

Modèle avec seulement une constante :

```
adf_const <- ur.df(PIB_2019, type = "drift", selectlags = "AIC")
summary(adf_const)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression drift

Call:

```
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-394420	-40546	9264	48733	196085

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.995e+04	3.942e+04	1.267	0.208
z.lag.1	4.857e-04	2.557e-03	0.190	0.850
z.diff.lag	3.669e-01	8.801e-02	4.169	6.04e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 82110 on 112 degrees of freedom

Multiple R-squared: 0.1361, Adjusted R-squared: 0.1207

F-statistic: 8.821 on 2 and 112 DF, p-value: 0.000277

Value of test-statistic is: 0.1899 13.633

Critical values for test statistics:

	1pct	5pct	10pct
tau2	-3.46	-2.88	-2.57
phi1	6.52	4.63	3.81

test-statistic : $0.1899 > \tau_2$, donc H_0 n'est pas rejeté, alors on continue avec le modèle suivant

Modèle sans tendance ni constante :

```
adf_none <- ur.df(PIB_2019, type = "none", selectlags = "AIC")
summary(adf_none)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression none

Call:

```
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
```

Residuals:

Min	1Q	Median	3Q	Max
-395692	-41726	9686	50361	191016

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
z.lag.1	0.0035983	0.0007122	5.052	1.69e-06 ***
z.diff.lag	0.3809072	0.0875482	4.351	2.99e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 82330 on 113 degrees of freedom

Multiple R-squared: 0.5757, Adjusted R-squared: 0.5682

F-statistic: 76.66 on 2 and 113 DF, p-value: $< 2.2e-16$

Value of test-statistic is: 5.0522

Critical values for test statistics:

```
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Test-statistic > tau1, alors H_0 est acceptée ici, la série est non stationnaire, il faudra donc différencier la série.

Que ce soit pas la méthode Dickey-Fuller simple ou augmentée, les test statistiques valident H_0 , alors la série du PIB est non stationnaire, il faudra différencier la série.

5. Appliquer le test de stationaritée de KPSS.

Hypothèse du test KPSS :

H (hypothèse nulle) : La série est stationnaire autour d'une moyenne ou d'une tendance linéaire.

H (hypothèse alternative) : La série est non-stationnaire.

```
kpss.test(PIB_2019)
```

KPSS Test for Level Stationarity

```
data:  PIB_2019
```

```
KPSS Level = 2.395, Truncation lag parameter = 4, p-value = 0.01
```

$p\text{-value} < 0.05$, alors on rejette H_0 , la série est non-stationnaire, le test de KPSS donne un résultat similaire aux test de Dickey-Fuller.

2. Période de 1990 à 2023

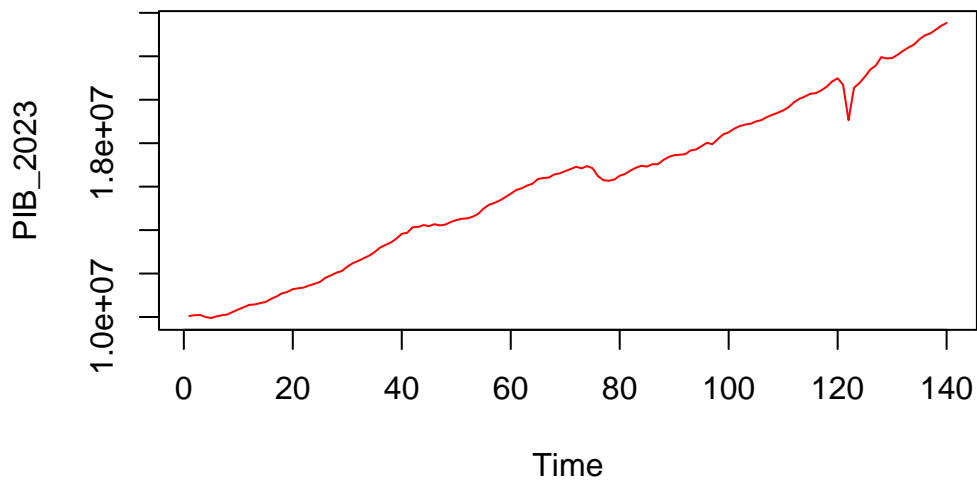
1. Mettre la série en time série et restreindre à la période 1990-2023

```
PIB_2023 <- ts(data = df_Y_us$PIB)
```

```
# Tracer la série temporelle
```

```
plot(PIB_2023, main="Time-Series du PIB des USA sur la période 1990 à 2023", col = "red")
```

Time-Series du PIB des USA sur la période 1990 à 2023



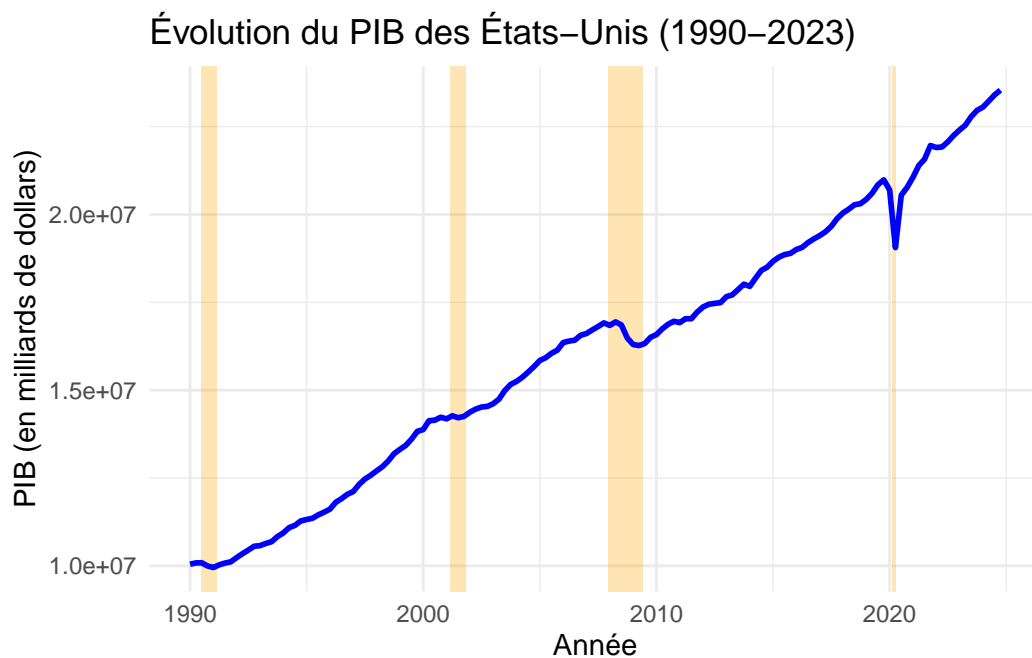
2. Zones ombrées pour les périodes de récession officielle aux US (1990-2023)

```
# Périodes de récession officielle aux US de 1990 à 2023
```

```
recessions <- data.frame(  
  start = as.Date(c("1990-07-01", "2001-03-01", "2007-12-01", "2020-02-01")),  
  end   = as.Date(c("1991-03-01", "2001-11-01", "2009-06-01", "2020-04-01"))  
)
```



```
# Tracer le graphique avec zones de récession
ggplot(df_Y_us, aes(x = Date, y = PIB)) +
  # Ajout des zones de récession
  geom_rect(data = recessions,
            aes(xmin = start, xmax = end, ymin = -Inf, ymax = Inf),
            fill = "orange", alpha = 0.3, inherit.aes = FALSE) +
  # Courbe du PIB
  geom_line(color = "blue", size = 1) +
  # Titres et labels
  labs(title = "Évolution du PIB des États-Unis (1990-2023)",
       x = "Année",
       y = "PIB (en milliards de dollars)") +
  # Thème épuré
  theme_minimal()
```

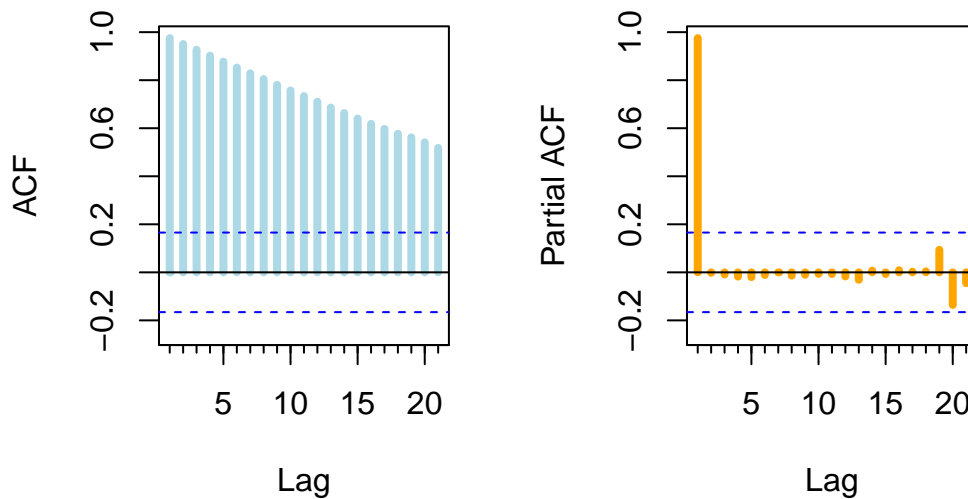


Les zones ombrées montrent bien les différents points de récessions officielle fourni par la National Bureau of Economic Research | NBER. La récession la plus marquante fut celle de 2008, qui dura le plus longtemps.

3. ACF et PACF de la série (avec la période 1990-2023)

```
par(mfrow=c(1,2))
Acf(PIB_2023, main = "ACF du PIB des USA (2023)", col="lightblue", lwd = 4)
Pacf(PIB_2023, main = "PACF du PIB des USA (2023)", col="orange", lwd = 4)
```

ACF du PIB des USA (2023) PACF du PIB des USA (2023)



```
par(mfrow=c(1,1))
```

La PACF montre que le PIB des USA dépend fortement du PIB de l'année précédente. L'ACF montre une décroissance lente et progressive des autocorrélations, cette décroissance graduelle sans coupure nette est caractéristique d'un processus non-stationnaire.

La série présente une forte tendance, ce qui est signe de non-stationnarité. Elle présente aussi une structure de dépendance où chaque observation est fortement liée aux observations précédentes et un comportement compatible avec un processus autorégressif d'ordre 1 (AR(1)) ou un modèle ARIMA avec composante intégrée.

Aucun retard significatif n'apparaît en prenant en compte la période globale.

4. Procédure de test de racine unitaire avec le test de Dickey-Fuller simple puis le test de Dickey-Fuller augmenté

a). Test de Dickey-Fuller Simple (DF) avec approche séquentielle

```
# Modèle avec tendance et constante
df_trend_const_2023 <- ur.df(PIB_2023, type = "trend", lags = 0)
summary(df_trend_const_2023)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression trend

Call:

```
lm(formula = z.diff ~ z.lag.1 + 1 + tt)
```

Residuals:

Min	1Q	Median	3Q	Max
-1767377	-49618	12110	63105	1179891

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.077e+06	3.955e+05	2.724	0.00730 **
z.lag.1	-1.047e-01	4.093e-02	-2.559	0.01158 *
tt	1.009e+04	3.813e+03	2.646	0.00909 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 209600 on 136 degrees of freedom

Multiple R-squared: 0.05133, Adjusted R-squared: 0.03738

F-statistic: 3.679 on 2 and 136 DF, p-value: 0.02779

Value of test-statistic is: -2.5593 12.3819 3.6792

Critical values for test statistics:

	1pct	5pct	10pct
tau3	-3.99	-3.43	-3.13

```
phi2  6.22  4.75  4.07
phi3  8.43  6.49  5.47
```

Comme la statistique du test de -2.5593 est supérieure à tau3 pour tous les niveaux de signification (1%, 5%, 10%), l'hypothèse nulle (H0 : série non stationnaire) n'est pas rejetée. La série est donc considérée comme non stationnaire, et nous devons la différencier pour obtenir une stationnarité.

```
# Modèle avec seulement une constante
df_const_2023 <- ur.df(PIB_2023, type = "drift", lags = 0)
summary(df_const_2023)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression drift

Call:

```
lm(formula = z.diff ~ z.lag.1 + 1)
```

Residuals:

Min	1Q	Median	3Q	Max
-1746668	-40407	15121	56574	1386765

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.144e+04	8.033e+04	0.640	0.523
z.lag.1	2.832e-03	4.860e-03	0.583	0.561

Residual standard error: 214200 on 137 degrees of freedom

Multiple R-squared: 0.002474, Adjusted R-squared: -0.004808

F-statistic: 0.3397 on 1 and 137 DF, p-value: 0.5609

Value of test-statistic is: 0.5829 14.4382

Critical values for test statistics:

	1pct	5pct	10pct
tau2	-3.46	-2.88	-2.57

```
phi1 6.52 4.63 3.81
```

La statistique de 0.5829 est bien supérieure aux valeurs critiques, et l'hypothèse nulle (présence de racine unitaire) ne peut pas être rejetée. La série reste non stationnaire.

```
# Modèle sans tendance ni constante
df_none_2023 <- ur.df(PIB_2023, type = "none", lags = 0)
summary(df_none_2023)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression none

Call:

```
lm(formula = z.diff ~ z.lag.1 - 1)
```

Residuals:

Min	1Q	Median	3Q	Max
-1757954	-28516	10619	53572	1380439

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
z.lag.1	0.005863	0.001097	5.347	3.62e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 213700 on 138 degrees of freedom

Multiple R-squared: 0.1716, Adjusted R-squared: 0.1656

F-statistic: 28.59 on 1 and 138 DF, p-value: 3.616e-07

Value of test-statistic is: 5.3468

Critical values for test statistics:

	1pct	5pct	10pct
tau1	-2.58	-1.95	-1.62

Le test-statistique de 5.3468 est bien supérieur à tau1, ce qui confirme que l'hypothèse nulle est validée. La série est non stationnaire et nécessite une différenciation.

b). Test de Dickey-Fuller Augmenté (ADF) avec approche séquentielle

```
# Modèle avec tendance et constante
adf_trend_const_2023 <- ur.df(PIB_2023, type = "trend", selectlags = "AIC")
summary(adf_trend_const_2023)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression trend

Call:

```
lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
```

Residuals:

Min	1Q	Median	3Q	Max
-1820767	-46867	17267	60228	974708

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	9.109e+05	4.094e+05	2.225	0.0277 *
z.lag.1	-8.647e-02	4.258e-02	-2.031	0.0443 *
tt	8.452e+03	3.960e+03	2.134	0.0347 *
z.diff.lag	-1.364e-01	8.694e-02	-1.569	0.1191

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 209300 on 134 degrees of freedom

Multiple R-squared: 0.06789, Adjusted R-squared: 0.04702

F-statistic: 3.253 on 3 and 134 DF, p-value: 0.02381

Value of test-statistic is: -2.0308 13.2023 2.5901

Critical values for test statistics:

	1pct	5pct	10pct
tau3	-3.99	-3.43	-3.13
phi2	6.22	4.75	4.07
phi3	8.43	6.49	5.47

La statistique de -2.0308 est inférieure à tau3 mais supérieure à tau1, ce qui indique que l'hypothèse nulle ne peut pas être rejetée à un niveau de signification de 1%. Cependant, la série reste non stationnaire.

```
# Modèle avec seulement une constante
adf_const_2023 <- ur.df(PIB_2023, type = "drift", selectlags = "AIC")
summary(adf_const_2023)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression drift

Call:

```
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
```

Residuals:

Min	1Q	Median	3Q	Max
-1823568	-32928	11499	59999	1062191

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.385e+04	8.047e+04	0.669	0.5045
z.lag.1	3.814e-03	4.884e-03	0.781	0.4362
z.diff.lag	-1.854e-01	8.495e-02	-2.183	0.0308 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 212000 on 135 degrees of freedom

Multiple R-squared: 0.03621, Adjusted R-squared: 0.02193

F-statistic: 2.536 on 2 and 135 DF, p-value: 0.08296

Value of test-statistic is: 0.7809 17.0766

Critical values for test statistics:

	1pct	5pct	10pct
tau2	-3.46	-2.88	-2.57
phi1	6.52	4.63	3.81

La statistique de 0.7809 est bien supérieure aux valeurs critiques pour tau2, et l'hypothèse nulle est acceptée. La série est non stationnaire et devra être différenciée.

```
# Modèle sans tendance ni constante
adf_none_2023 <- ur.df(PIB_2023, type = "none", selectlags = "AIC")
summary(adf_none_2023)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression none

Call:

```
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
```

Residuals:

Min	1Q	Median	3Q	Max
-1835282	-29475	14184	60053	1055597

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
z.lag.1	0.006981	0.001200	5.817	4.08e-08 ***
z.diff.lag	-0.185467	0.084777	-2.188	0.0304 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 211600 on 136 degrees of freedom

Multiple R-squared: 0.1997, Adjusted R-squared: 0.1879

F-statistic: 16.97 on 2 and 136 DF, p-value: 2.638e-07

Value of test-statistic is: 5.8175

Critical values for test statistics:

	1pct	5pct	10pct
tau1	-2.58	-1.95	-1.62

Le test-statistique de 5.8175 est bien supérieur aux valeurs critiques pour tau1, ce qui confirme que l'hypothèse nulle de racine unitaire est validée. La série est non stationnaire.

5. Appliquer le test de stationnarité de KPSS (pour la période 1990-2023)

```
# Test de KPSS  
kpss.test(PIB_2023)
```

KPSS Test for Level Stationarity

```
data:  PIB_2023  
KPSS Level = 2.8352, Truncation lag parameter = 4, p-value = 0.01
```

Le p-value est inférieur à 0.05, ce qui permet de rejeter l'hypothèse nulle (stationnarité autour de la moyenne ou d'une tendance linéaire). Cela confirme également que la série est non stationnaire.

Les résultats des tests de Dickey-Fuller et du test KPSS montrent que la série du PIB 2023 est non stationnaire. En effet, tant les tests DF que l'ADF n'ont pas permis de rejeter l'hypothèse nulle, et le test KPSS a également rejeté l'hypothèse de stationnarité. La série du PIB doit donc être différenciée pour obtenir une série stationnaire.

3. Conclusion

Les conclusions principales concernant la non-stationnarité de la série du PIB des États-Unis ne sont pas modifiées par l'extension de la période d'analyse de 1990 à 2023. L'ajout de la crise de 2008 et de la récession liée à la pandémie de COVID-19 en 2020 renforce plutôt l'idée que la série du PIB présente des tendances non stationnaires sur une période plus longue, ce qui est confirmé par tous les tests. La série du PIB reste fortement liée à ses valeurs passées, ce qui nécessite une différenciation pour obtenir une stationnarité.

Ainsi, bien que les périodes de récession supplémentaires apportent un contexte enrichi, la conclusion fondamentale — que la série est non stationnaire et nécessite des ajustements — reste inchangée.