

INGENIEUR STATISTICIEN DATA –  
ANALYST ET DATA- SCIENCE

**SERIE TEMPORELLE**

**PREVISIONS TRENTE  
DERNIERS JOURS  
POLLUTION PEKIN**

**NOM DU PROFESSEUR**

**MR. AKPOSSO DIDIER  
MARTIAL**

**NOM DE L'ETUDIANTE**

**KOUAHON ESTELLE**

## AVANT-PROPOS

La qualité de l'air est une préoccupation majeure de santé publique et environnementale à l'échelle mondiale. La surveillance et la prévision des niveaux de pollution atmosphérique sont essentielles pour informer les populations, mettre en œuvre des politiques de réduction des émissions et anticiper les épisodes de forte pollution. Les données de pollution, souvent collectées quotidiennement, se présentent naturellement sous forme de séries temporelles, caractérisées par des tendances, des saisonnalités et des volatilités spécifiques. Le jeu de données utilisé pour cette étude provient de la ville de Pékin, récupéré sur la plateforme Kaggle, et a été fourni par Monsieur Akposso Martial, Directeur des Études, dans le cadre d'une évaluation de nos capacités à réaliser des prévisions en séries temporelles avec des méthodes avancées (SARIMA, ARIMA, GARCH).

La prévision précise des niveaux de pollution futurs représente un défi en raison de la complexité inhérente aux séries temporelles environnementales. Ces séries peuvent présenter des valeurs extrêmes, une variance non constante au fil du temps, ainsi que des motifs saisonniers multiples (hebdomadaires, annuels) qui doivent être correctement identifiés et modélisés pour garantir la fiabilité des prévisions. L'objectif est de développer un modèle capable de capturer ces dynamiques pour anticiper les niveaux de pollution et ainsi contribuer à l'élaboration d'initiatives de prévention spécifiques à Pékin.

L'objectif général de cette étude est de développer un modèle de prévision robuste et fiable des niveaux de pollution atmosphérique à partir de données historiques, afin de fournir des estimations précises pour les jours à venir. Pour atteindre cet objectif général, plusieurs étapes spécifiques ont été entreprises. Assurer la qualité et la structure adéquate des données historiques de pollution, y compris le traitement des valeurs manquantes, des doublons et des valeurs extrêmes. Stabiliser la variance de la série temporelle pour répondre aux hypothèses des modèles statistiques. Décomposer la série pour identifier et quantifier la tendance et les différentes composantes saisonnières (hebdomadaire et annuelle). Appliquer la méthodologie de Box & Jenkins pour spécifier, estimer et valider un modèle SARIMA (Seasonal Autoregressive Integrated Moving Average) capable de capturer les dynamiques

identifiées. Evaluer rigoureusement la pertinence du modèle par l'analyse des résidus et des métriques de performance sur un jeu de test. Utiliser le modèle validé pour générer des prévisions fiables des niveaux de pollution pour les 30 prochains jours, présentées à l'échelle originale et accompagnées d'intervalles de confiance.

Pour atteindre ces objectifs, une démarche rigoureuse de modélisation de séries temporelles, inspirée de la méthodologie de Box & Jenkins, a été adoptée. Elle a impliqué des phases successives de préparation et d'exploration des données, de stationnarisation et d'identification des ordres du modèle, d'estimation et d'itérations pour affiner le modèle, et enfin de génération et de présentation des prévisions. Les détails de ce plan seront exposés dans la section suivante pour une compréhension exhaustive des étapes entreprises.

# SOMMAIRE

## Sommaire

<b>1. Introduction</b>	1
• Contexte	1
• Problématique	1
• Objectif général et objectifs spécifiques	2
• Résultats attendus	3
• Démarche et justification méthodologique	4
<b>2. Prétraitement des données</b>	6
• Importation et description des données	6
• Gestion des valeurs manquantes et des doublons	7
• Nettoyage et préparation du jeu final	8
<b>3. Analyse exploratoire des données (EDA)</b>	9
• Analyse statistique univariée	9
• Analyse statistique bivariée	10
• Analyse de la variance	11
• Transformation logarithmique	12
• Analyse saisonnière (mois, année, chronogramme)	13
• Décomposition STL	14
<b>4. Modélisation et prévisions</b>	15
• Stationnarisation et identification des ordres	15
• Présentation des modèles testés	17
• Comparaison des performances	20
• Sélection du modèle optimal	22
• Prévisions sur les 30 derniers jours	23
<b>5. Discussion</b>	25
• Interprétation des résultats	25
• Limites	26
• Recommandations	27
<b>6. Conclusion</b>	40

## Codes Sources

## INTRODUCTION

- **Contexte**

La qualité de l'air constitue une préoccupation environnementale et sanitaire croissante à l'échelle mondiale. L'Organisation Mondiale de la Santé (OMS) considère la pollution de l'air comme la première menace environnementale pour la santé humaine, responsable de **7 millions de décès prématurés chaque année**. Parmi les différentes formes de pollution, les **particules fines PM2.5** (d'un diamètre inférieur à 2,5 microns) sont particulièrement dangereuses car elles pénètrent profondément dans les voies respiratoires et peuvent entrer dans la circulation sanguine, provoquant des maladies cardiovasculaires, respiratoires et des troubles neurologiques.

La ville de **Pékin**, capitale de la Chine, est l'une des métropoles les plus exposées à la pollution atmosphérique, en particulier pendant les mois d'hiver. Cette pollution provient majoritairement de **quatre sources principales** :

- La **combustion du charbon** pour le chauffage résidentiel,
- Les **émissions industrielles**,
- La **circulation automobile** intense,
- Les **tempêtes de poussière** liées au climat aride et à la désertification.

En hiver, les niveaux de **PM2.5 peuvent dépasser les 250  $\mu\text{g}/\text{m}^3$** , soit **plus de 15 fois la limite recommandée par l'OMS** (15  $\mu\text{g}/\text{m}^3$  sur 24h), entraînant des **alertes rouges** et des fermetures temporaires d'écoles et d'usines.

La surveillance et la prévision de cette pollution sont devenues des enjeux clés pour permettre aux autorités d'**anticiper les pics**, de **protéger les populations vulnérables** (enfants, personnes âgées, femmes enceintes) et de **mettre en œuvre des politiques de réduction des émissions**.

- **Problématique**

La prévision des niveaux futurs de pollution est une tâche complexe car les séries temporelles environnementales présentent :

- Une **variance instable dans le temps** (surtout entre saisons),
- Des **saisonnalités multiples** (hebdomadaire : activité humaine ; annuelle : hiver/été),
- Des **valeurs extrêmes et des anomalies locales** (ex. : événements météo ou fêtes avec feux d'artifice),
- Une dépendance potentielle à **des variables exogènes** comme la météo (température, pression, vent, pluie).

L'objectif est donc de développer un modèle capable de **capturer ces dynamiques complexes** et de fournir des prévisions suffisamment fiables pour un usage opérationnel ou scientifique.

- **Objectif général**

L'objectif général de cette étude est de **développer un modèle robuste et interprétable de prévision des niveaux quotidiens de pollution atmosphérique à Pékin**, en s'appuyant sur l'analyse d'une série temporelle issue de données historiques.

- **Objectifs spécifiques**

Afin d'atteindre cet objectif, plusieurs sous-objectifs ont été poursuivis :

- **Préparation et exploration des données** : Assurer la qualité du jeu de données, détecter et traiter les valeurs manquantes et aberrantes.
- **Transformation de la série** : Stabiliser la variance et extraire les composantes (tendance, saisonnalités).
- **Stationnarisation et identification** : Rendre la série stationnaire et identifier les ordres du modèle.
- **Modélisation SARIMA multisaisonnière** : Appliquer la méthode Box-Jenkins pour modéliser les composantes détectées.
- **Diagnostic du modèle** : Analyser les résidus, tester le caractère de bruit blanc, vérifier les hypothèses.
- **Prévision et validation** : Générer des prévisions fiables sur les **30 derniers jours du jeu de données**, en les retranscrivant à l'échelle d'origine pour une interprétation réaliste, et les comparer aux valeurs réelles de la même période à l'aide de métriques classiques (MAPE, RMSE, MAE).

- **Démarche et justification méthodologique**

La méthodologie adoptée suit le schéma de Box & Jenkins, enrichi par l'utilisation de modèles alternatifs et multivariés.

1. **Préparation et exploration des données**

- Nettoyage des données `air_final` : suppression des variables non significatives (`snow`, `rain`, `weekday`).
- Analyse statistique univariée et bivariée, visualisation des tendances, saisonnalités et anomalies.

2. **Transformation et stationnarisation**

- Application d'une transformation logarithmique stabilisante : `pollution_log <- log(pollution_today + 1)`.
- Identification et suppression de la tendance et des effets saisonniers par différenciation.

3. **Création des variables explicatives**

- Variables météorologiques : température (`temp`), pression (`press`), vitesse du vent (`wnd_spd`).
- Variables issues de la pollution passée : `pollution_yesterday` (lag 1), moyennes mobiles (MA3, MA7).
- Termes quadratiques ( $temp^2$ ,  $press^2$ ,  $wnd\_spd^2$ ) pour capter des effets non linéaires.
- Termes de Fourier pour les saisonnalités multiples (hebdomadaire, annuelle).

4. **Modélisation**

- Tests de différents modèles :
  - **TBATS seul** (multi-saisonnalités complexes, pas d'exogènes),
  - **Prophet** (tendance + saisonnalités annuelles et hebdomadaires),
  - **TBATS + exogènes** (météo),
  - **SARIMAX simple** (météo uniquement),
  - **SARIMAX enrichi** (météo + pollution passée + effets quadratiques + Fourier).
- Choix final : **SARIMAX enrichi** pour son meilleur compromis précision / interprétabilité (RMSE = 58.78, MAPE = 60.46%).

5. **Validation**

- Division des données en ensemble d'entraînement et de test (30 derniers jours pour test).
- Évaluation via RMSE, MAPE, MAE.
- Comparaison visuelle des prévisions et observations avec intervalles de confiance.

## 6. **Interprétation et recommandations**

- Analyse des performances, identification des limites, formulation de pistes d'amélioration.

### • **Résultats Attendus**

À l'issue de cette démarche, il est attendu :

- Un **modèle statistiquement valide**, avec des résidus indépendants et homoscedastiques,
- Des **prévisions fiables sur les 30 derniers jours** (période de test), avec une précision satisfaisante,
- Une **interprétation métier des prévisions** : mise en évidence des périodes critiques, notamment les pics hivernaux attendus à Pékin,
- Des **recommandations opérationnelles** pour l'usage de ces prévisions par des autorités publiques, organismes de santé ou citoyens concernés.

### • **Limites et Recommandations**

Enfin, nous proposerons une discussion des **limites de la modélisation** : nature univariée initiale, qualité des données météo, absence d'événements exogènes comme les fêtes, etc.

Nous développerons également des **recommandations concrètes** :

- Vers l'**intégration de sources de données en temps réel**,
- L'**amélioration de la granularité spatiale (par quartier)**,
- L'usage de **modèles hybrides IA/statistiques**.

Enfin, nous concluons sur l'importance d'une telle modélisation pour **protéger la santé publique**, en particulier à Pékin, où la pollution est un problème structurel, mais potentiellement prévisible et donc atténuable.



- **Dictionnaire De Données**

Le dictionnaire de données fournit une documentation exhaustive de chaque variable présente dans le jeu de données, détaillant son nom, son type, sa description, et son unité de mesure.

Nom de la variable	Type données (initial)	Description	Unité de mesure	Note complémentaire
Date	Factor	Date de l'observation. Initialement importée comme un facteur, elle a été convertie au format Date pour permettre les opérations chronologiques.	Date (AAAA-MM-JJ)	Couvre une période de 1825 jours (du 2 janvier 2010 au 31 décembre 2014) dans le dataset utilisé, bien que la description initiale mentionne 2014-2019.
<b>pollution_today</b>	Numérique	Température du point de rosée. C'est la température à laquelle l'air doit être refroidi pour que la vapeur d'eau qu'il contient atteigne la saturation et condense.	Degrés Celsius (°C)	Indicateur d'humidité.
<b>temp</b>	Numérique	Température de l'air ambiant.	Degrés Celsius (°C)	Variable météorologique clé pouvant influencer la dispersion des polluants.

<b>Press</b>	Numérique	Pression atmosphérique.	Hectopascals (hPa)	Une pression élevée peut indiquer des conditions stables favorisant l'accumulation de polluants.
<b>wnd_spd</b>	Numérique	Vitesse du vent.	Non spécifiée (ex: m/s, km/h)	La vitesse du vent est cruciale pour la dispersion des polluants. Des valeurs extrêmes ont été identifiées et traitées par winsorisation pour cette variable.
<b>Snow</b>	Numérique	Quantité de neige enregistrée.	Millimètres (mm)	Valeur de 0 mm indique l'absence de neige.
<b>Rain</b>	Numérique	Quantité de pluie enregistrée.	Millimètres (mm)	Valeur de 0 mm indique l'absence de pluie. Les précipitations peuvent aider à "laver" l'atmosphère des polluants.
<b>pollution_yesterday</b>	Numérique	Niveau de pollution de l'air mesuré pour la veille (date - 1). Cette variable sert de prédicteur potentiel ou de référence pour	Non spécifiée (ex: AQI, PM2.5)	Permet de capturer la dépendance des niveaux de pollution actuels vis-à-vis des niveaux passés, souvent une caractéristique des séries temporelles de pollution. Exporter vers Sheets

		l'inertie de la pollution.		

## PRETRAITEMENT DES DONNEES

Cette première partie du projet est consacrée à la phase cruciale de préparation et d'exploration des données. Avant toute tentative de modélisation prédictive, il est impératif d'assurer la qualité, la propreté et la bonne structuration du jeu de données. Cette étape initiale permet de comprendre les caractéristiques fondamentales des données, d'identifier d'éventuels problèmes (valeurs manquantes, doublons, valeurs extrêmes) et de procéder aux transformations nécessaires pour rendre la série temporelle apte à l'analyse statistique. Une exploration approfondie des composants de la série (tendance, saisonnalités) est également menée pour éclairer les choix de modélisation futurs.

### 1- Importation et Vérification Initiale des Données

Le jeu de données `air_pollution.csv`, provenant de la plateforme Kaggle et fourni par Monsieur Akposso Martial, Directeur des Études de l' INSEDS ( Institut Nationale de Statistique et de Data Science), dans le cadre d'une évaluation de nos capacités à réaliser des prévisions en séries temporelles avec des méthodes avancées (SARIMA, ARIMA, GARCH). Ce jeu de données a été renommé « air » pour des raisons évidentes. Il a été chargé dans l'environnement R via l'opération `air <- read.csv("air_pollution.csv", stringsAsFactors=TRUE)`. Ce dataset, issu de la ville de Pékin, contient des données sur la qualité de l'air et des caractéristiques météorologiques (pression, température, etc.), initialement échantillonnées toutes les 10 minutes puis consolidées quotidiennement.

	<b>date</b> <fctr>	<b>pollution_today</b> <dbl>	<b>dew</b> <dbl>	<b>temp</b> <dbl>	<b>press</b> <dbl> ▶
1	2010-01-02	145.95833	-8.50000	-5.125000	1024.750
2	2010-01-03	78.83333	-10.12500	-8.541667	1022.792
3	2010-01-04	31.33333	-20.87500	-11.500000	1029.292
4	2010-01-05	42.45833	-24.58333	-14.458333	1033.625
5	2010-01-06	56.41667	-23.70833	-12.541667	1033.750
6	2010-01-07	69.00000	-21.25000	-12.500000	1034.083

Un aperçu des premières lignes (`head(air)`) a confirmé le bon chargement et la présence attendue des colonnes `date`, `pollution_today`, `pollution_yesterday`, `dew`, `temp`, `press`, `wnd_spd`, `snow`, et `rain`. Les variables `pollution_today` et `pollution_yesterday` représentent les niveaux de pollution

pour le jour courant et le jour précédent, respectivement, dans le but de prédire ces niveaux et de prendre des initiatives de prévention.

- Vérification de la Structure (str(air))

L'analyse de la structure du dataset a révélé que la colonne date était initialement importée comme un facteur. Les autres variables étaient correctement identifiées comme des types numériques (dbl).

```
'data.frame': 1825 obs. of 9 variables:
 $ date      : Factor w/ 1825 levels "2010-01-02","2010-01-03",...: 1 2 3 4 5 6 7 8 9 10 ...
 $ pollution_today : num 146 78.8 31.3 42.5 56.4 ...
 $ dew        : num -8.5 -10.1 -20.9 -24.6 -23.7 ...
 $ temp       : num -5.12 -8.54 -11.5 -14.46 -12.54 ...
 $ press      : num 1025 1023 1029 1034 1034 ...
 $ wnd_spd    : num 24.9 70.9 111.2 56.9 18.5 ...
 $ snow       : num 0.708 14.167 0 0 0 ...
 $ rain       : num 0 0 0 0 0 0 0 0 0 0 ...
 $ pollution_yesterday: num 10 146 78.8 31.3 42.5 ...
[1] 1825 9
```

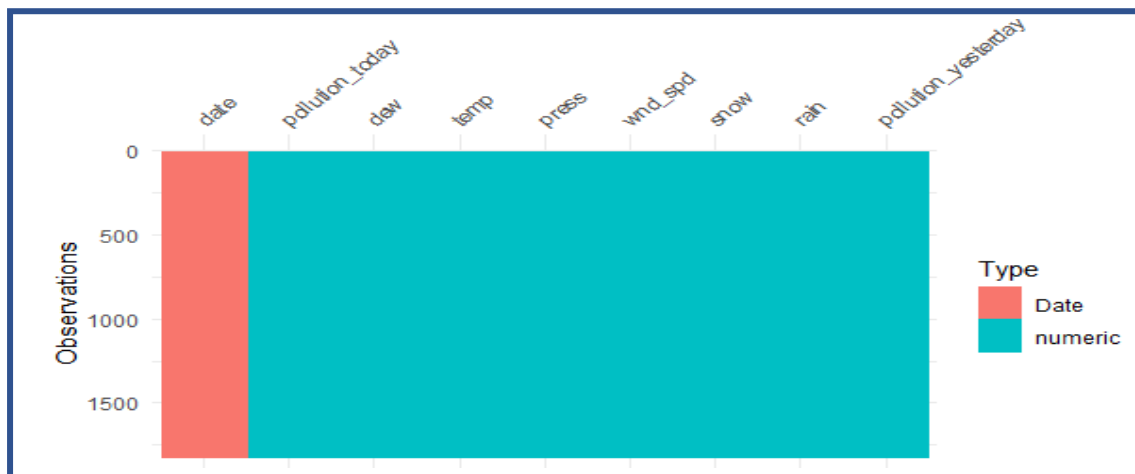
- Conversion de la colonne Date en format Date

Une étape fondamentale dans le prétraitement des données de séries temporelles est la conversion correcte de la colonne temporelle. Initialement importée en tant que facteur (ou chaîne de caractères), la colonne date a été explicitement transformée en un objet de type Date en utilisant la fonction as.Date() de R et en spécifiant le format "%Y-%m-%d".

```
'data.frame': 1825 obs. of 9 variables:
 $ date      : Date, format: "2010-01-02" ...
 $ pollution_today : num 146 78.8 31.3 42.5 56.4 ...
 $ dew        : num -8.5 -10.1 -20.9 -24.6 -23.7 ...
 $ temp       : num -5.12 -8.54 -11.5 -14.46 -12.54 ...
 $ press      : num 1025 1023 1029 1034 1034 ...
 $ wnd_spd    : num 24.9 70.9 111.2 56.9 18.5 ...
 $ snow       : num 0.708 14.167 0 0 0 ...
 $ rain       : num 0 0 0 0 0 0 0 0 0 ...
 $ pollution_yesterday: num 10 146 78.8 31.3 42.5 ...
```

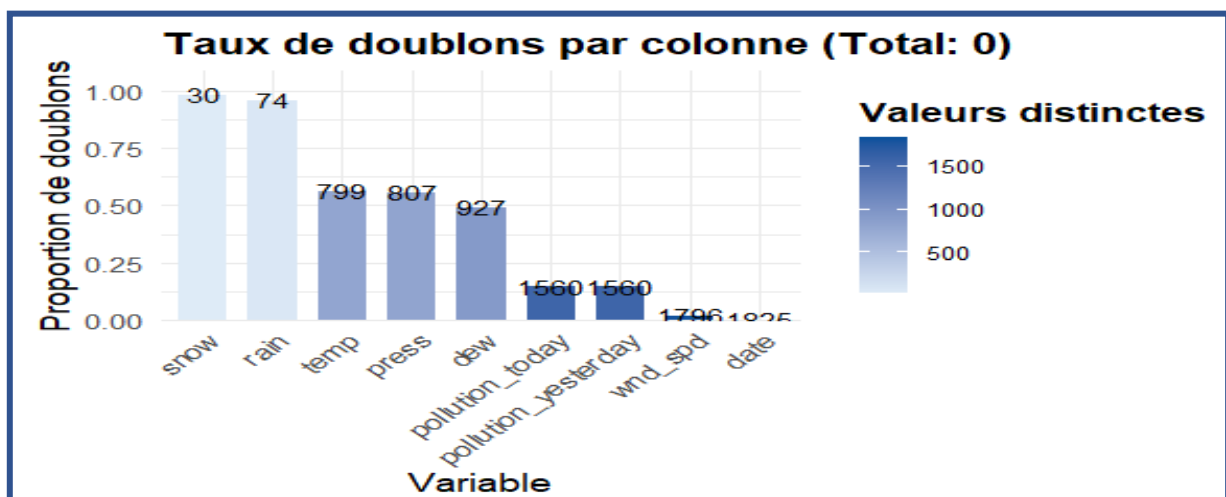
- Vérification de l'Absence de Valeurs Manquantes.

Des vérifications exhaustives ont été menées pour s'assurer de l'intégrité du jeu de données. Le graphique de complétude (souvent généré par des packages d'EDA) ainsi que la fonction `colSums(is.na(air))` ont unanimement confirmé **l'absence totale de valeurs manquantes** dans toutes les colonnes du dataset. Cette absence simplifie grandement les étapes ultérieures de prétraitement.



- Vérification de l'Absence de Doublons.

La commande `sum(duplicated(air))` a retourné la valeur 0, indiquant qu'**aucune ligne en double** n'était présente dans le jeu de données. Ceci garantit que chaque observation est unique et représente une période distincte, une condition essentielle pour l'analyse des séries temporelles.



## 2- Gestion de l'Index Temporel

### ○ Conversion de la colonne date au format Date.

Une étape fondamentale dans le prétraitement des données de séries temporelles est la conversion correcte de la colonne temporelle. Initialement importée en tant que facteur (ou chaîne de caractères), la colonne date a été explicitement transformée en un objet de type Date en utilisant la fonction `as.Date()` de R et en spécifiant le format "%Y-%m-%d".

### ○ Vérification de la continuité et de la complétude de la série de dates

La vérification de l'exhaustivité des dates est une étape cruciale qui vise à s'assurer de la régularité et de la complétude de l'index temporel lui-même. Le principe est de comparer la série de dates présente dans les données avec une série de dates "idéale" qui devrait exister pour la période d'étude, à la fréquence attendue (quotidienne ici). Si les longueurs de ces deux séries de dates sont identiques, cela confirme qu'il n'y a ni dates manquantes ni doublons dans la séquence temporelle. Cette vérification est distincte de la recherche de valeurs manquantes (NA) ou atypiques au sein des observations des variables.

- **Vérification de la Période Couverte :** Les dates minimale et maximale du dataset (`min(air$date)` et `max(air$date)`) ont été inspectées, révélant une période d'étude s'étendant du **2 janvier 2010 au 31 décembre 2014**.
- **Vérification de la Continuité :** La continuité de la série de dates a été confirmée, assurant qu'il n'y avait pas de jours manquants au sein de la période couverte. Cette confirmation s'appuie sur le principe d'exhaustivité, où la comparaison entre le nombre d'observations et le nombre de jours attendus sur la période (1825 jours du 2010-01-02 au 2014-12-31) a validé l'absence de lacunes.

✓ La longueur de date traitée est la même que celle de la série générée.  
🎯 La variable date traitée est parfaitement traitée.

### 3- Traitement des valeurs aberrantes et extrêmes

Dans l'analyse des séries temporelles, les valeurs aberrantes sont des observations qui dévient significativement de la norme. Leur présence peut gravement fausser l'estimation des modèles et réduire la précision des prévisions, car elles biaisent les calculs statistiques et les dynamiques identifiées. Il est donc crucial de les identifier et de les traiter adéquatement pour garantir la robustesse et la fiabilité de notre modèle de prévision.

#### ○ Visualisation avant traitement des valeurs aberrantes et extrêmes

Les boîtes à moustaches (boxplots) confirment les observations du summary().

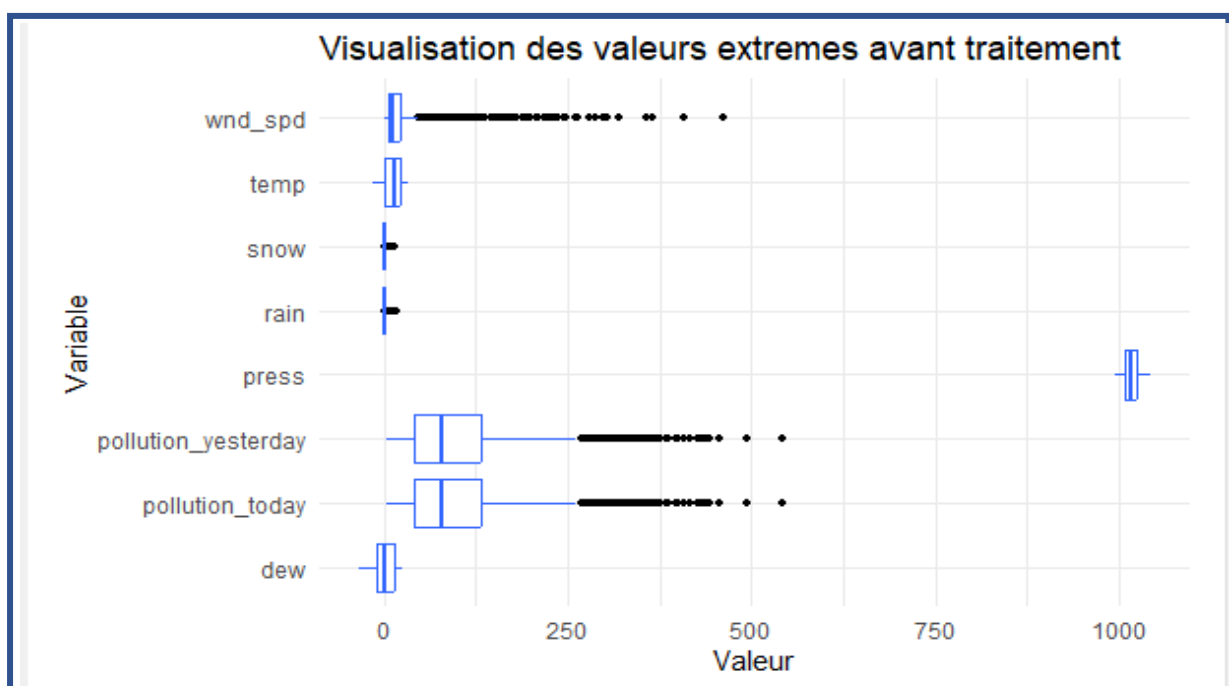
wnd\_spd : Présente de très nombreux points au-delà de la moustache supérieure, indiquant un grand nombre de valeurs extrêmes (outliers) ou une distribution fortement asymétrique. La valeur maximale à 463 est clairement visible.

snow et rain : Beaucoup de zéros, avec quelques points isolés pour les jours de précipitations.

press : Semble avoir une distribution très serrée avec quelques valeurs aberrantes à l'extrémité inférieure.

pollution\_today et pollution\_yesterday : Présentent également des points au-delà de la moustache supérieure, indiquant des épisodes de forte pollution.

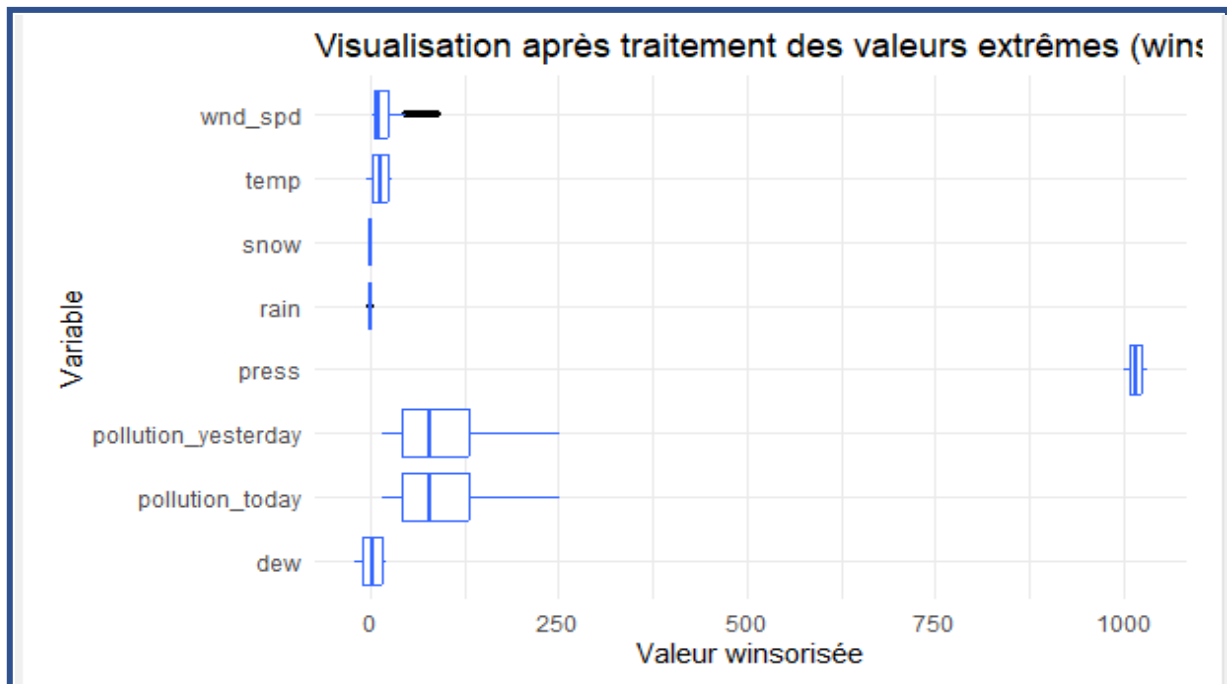
dew et temp : Semblent avoir des distributions plus symétriques mais avec des valeurs extrêmes aux deux extrémités, ce qui est normal pour des températures.





### ○ Visualisation après traitement des valeurs aberrantes et extrêmes

L'application de la winsorisation a visiblement réduit l'impact de ces valeurs extrêmes. Les boxplots montrent des moustaches plus courtes et une diminution notable des points isolés, indiquant que les valeurs aberrantes de `wnd_spd`, `pollution_today`, `pollution_yesterday` et `press` ont été ramenées à des limites prédéfinies, rendant les distributions plus robustes.



### ○ Conclusion partielle

L'analyse exploratoire a permis de nettoyer et préparer efficacement le jeu de données.

Les statistiques univariées et bivariées ont mis en évidence :

- Une forte variabilité saisonnière, avec des pics hivernaux marqués.
- Une variance instable entre saisons, justifiant une transformation logarithmique (`log1p`) pour stabiliser l'échelle.
- Des corrélations significatives entre pollution et variables météorologiques (température, pression, vitesse du vent).

Les variables `snow`, `rain` et `weekday` ont été supprimées car elles n'apportaient pas de valeur prédictive significative dans ce contexte.

La décomposition STL et les chronogrammes ont confirmé la présence de tendances et de saisonnalités multiples (hebdomadaire et annuelle), orientant vers un modèle

SARIMAX enrichi intégrant à la fois des effets météorologiques et des variables dérivées (lags, moyennes mobiles, termes quadratiques).

## ANALYSE EXPLORATOIRE DES DONNEES (EDA)

Cette deuxième phase du projet vise à explorer en profondeur la série temporelle des niveaux de pollution journaliers à Pékin, ainsi que ses éventuelles dépendances vis-à-vis des facteurs météorologiques. L'analyse exploratoire (EDA) permet de détecter les structures cachées dans les données : **tendance de long terme, fluctuations saisonnières, instabilités de variance**, mais aussi **corrélations et relations entre les variables**.

Dans un contexte de pollution atmosphérique, cette étape est essentielle pour :

- **Comprendre les cycles temporels spécifiques** (ex. : pics hivernaux, effets hebdomadaires),
- **Vérifier les hypothèses de stationnarité** requises pour les modèles temporels,
- **Déterminer si les conditions météorologiques expliquent partiellement les variations de pollution.**

L'analyse débute par des **statistiques univariées** (distribution, dispersion, asymétrie), suivies de **statistiques bivariées** pour explorer les dépendances entre pollution et météo. Elle se prolonge avec l'étude de la **variance locale**, la **transformation logarithmique**, la **visualisation de la tendance** et la **décomposition STL**, permettant d'isoler les composantes saisonnières.

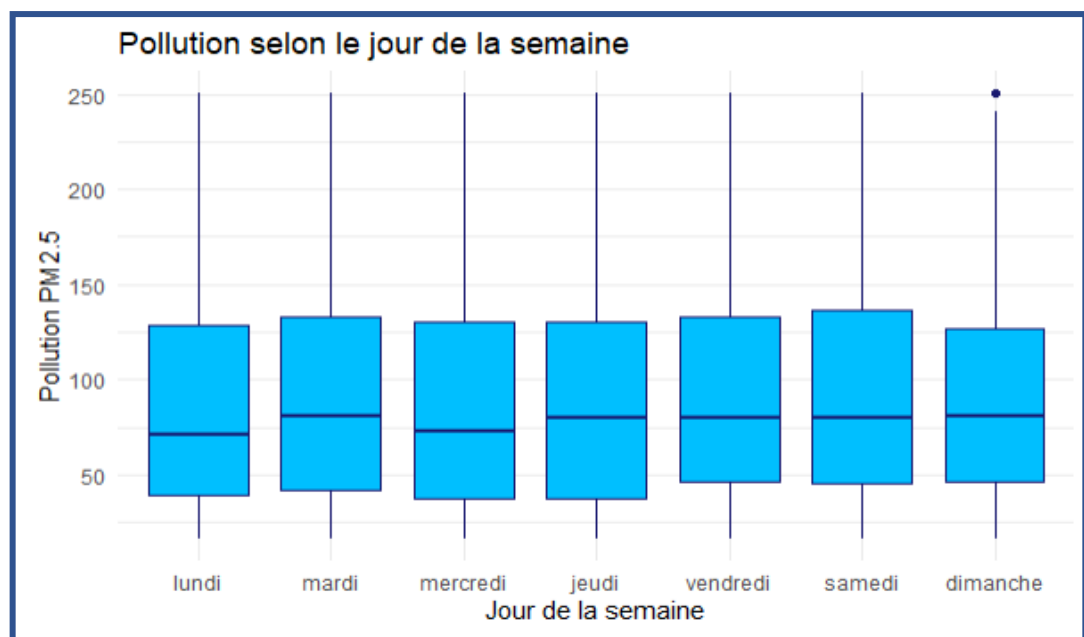
### 1- Résumés Global Descriptif

	pollution_today	dew	temp	Press	wnd_spd	snow	rain	pollution_yes
Moyenne	94.72	1.88	12.51	1016	19.62	0.00	0.081	94.72
Médiane	79.17	2.04	13.91	1016	10.96	0.00	0.00	79.17
Écart-type	65.37	13.86	11.28	9.66	22.24	0.00	0.22	65.37
Min	16.52	-20.74	-5.49	1001	2.64	0.00	0.00	16.52
Max	250.28	22.08	27.91	1033	88.06	0.00	0.87	250.28

○ Interprétations

Variable	Interprétation
<i>pollution_today</i>	La moyenne est très élevée ( <b>94,7 <math>\mu\text{g}/\text{m}^3</math></b> ), avec des pics atteignant <b>250 <math>\mu\text{g}/\text{m}^3</math></b> , bien au-dessus du seuil OMS ( <b>15 <math>\mu\text{g}/\text{m}^3</math></b> ). Cela reflète de <b>forts épisodes de pollution</b> , surtout en hiver.
<i>Dew</i>	Moyenne proche de celle de la température ( <b>1,9°C</b> ), avec une forte variabilité ( <b><math>\sigma = 13,9</math></b> ). Des valeurs très basses sont fréquentes en hiver, favorisant l'air sec et la <b>stagnation des polluants</b> .
<i>Temp</i>	Températures allant de <b>-5,5°C à 28°C</b> , avec une moyenne de <b>12,5°C</b> . Le froid hivernal joue un rôle clé dans <b>l'accumulation de pollution</b> , à cause du chauffage et des inversions thermiques.
<i>Press</i>	Moyenne normale ( <b>1016 hPa</b> ), mais de petites variations peuvent <b>influencer la stabilité de l'air</b> et limiter la dispersion de la pollution.
<i>wnd_spd</i>	Moyenne de <b>19,6 m/s</b> , mais forte dispersion ( <b><math>\sigma = 22,2</math></b> ). Le vent est un <b>facteur crucial de dilution</b> : faible vent = pollution plus persistante.
<i>Snow</i>	Toujours nulle : <b>aucun événement neigeux mesuré</b> sur la période → cette variable peut être ignorée.
<i>Rain</i>	Moyenne très faible ( <b>0,08 mm</b> ) : la pluie est rare, mais peut ponctuellement <b>nettoyer l'air</b> en cas d'épisode.
<i>pollution_yesterday</i>	Même distribution que <i>pollution_today</i> , avec un décalage temporel : servira comme <b>variable autoregressive</b> dans le modèle

○ Graphique de la distribution mensuelle de la pollution



On remarque que les niveaux moyens de pollution varient fortement selon les mois de l'année.

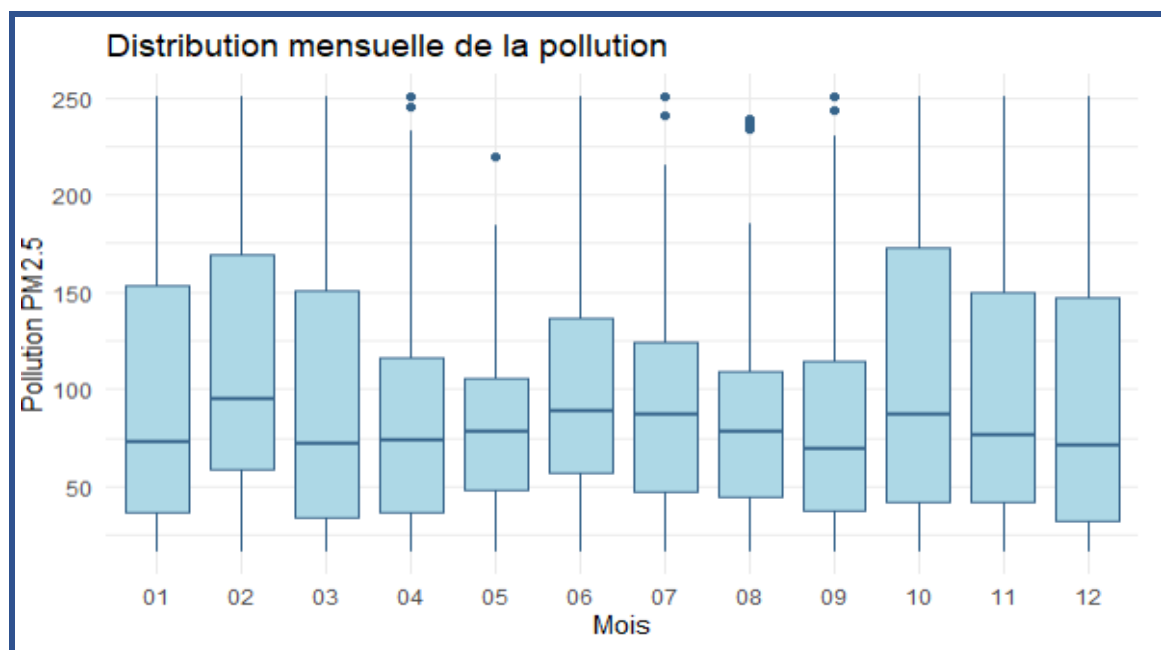
Les pics se situent principalement en **hiver** (décembre, janvier, février), ce qui peut s'expliquer par une combinaison de conditions météorologiques défavorables (inversions thermiques) et une augmentation de l'utilisation du chauffage au charbon. ( **Mois les plus pollués**)

L'été (juin à août) présente généralement des niveaux plus bas, possiblement grâce à une meilleure dispersion des particules due à des vents plus forts et à la convection thermique.

(**Mois les moins pollués**)

Ces variations saisonnières devront être intégrées dans la modélisation (via des termes saisonniers ou de Fourier), car elles influencent significativement la concentration de particules.

#### ○ Distribution hebdomadaire de la pollution



La pollution varie également selon le jour de la semaine, bien que les écarts soient moins marqués que la variation mensuelle.

**Tendance générale :**

- Légère hausse en **milieu de semaine** (mardi à jeudi), ce qui peut être lié à l'intensité du trafic et des activités industrielles.

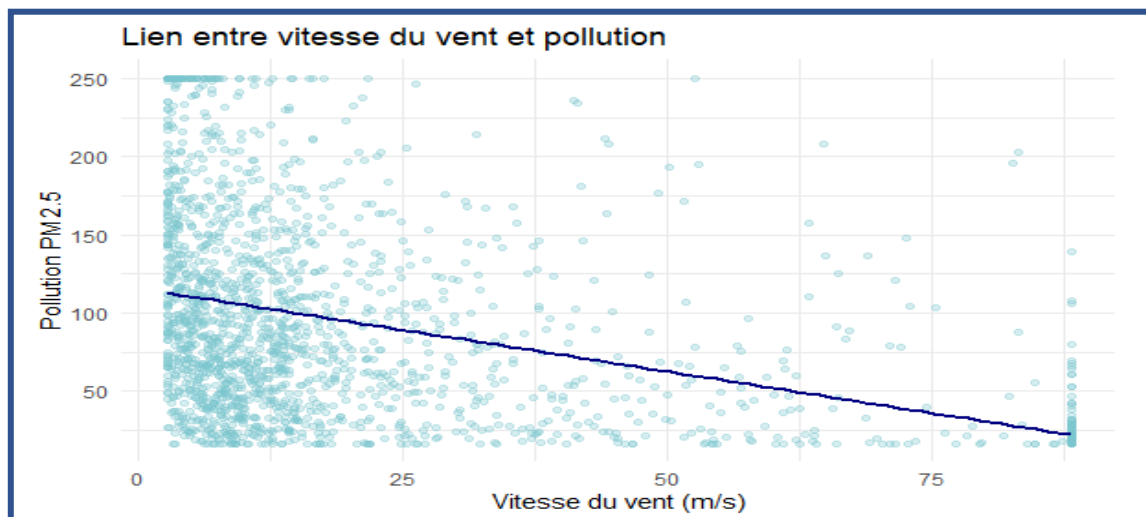
- Niveaux plus faibles en **week-end** (samedi et dimanche), probablement en raison d'une diminution des déplacements et de l'activité industrielle.

Même si l'effet hebdomadaire est plus subtil que l'effet saisonnier annuel, il peut contribuer à améliorer les performances des modèles lorsqu'on inclut une composante de saisonnalité hebdomadaire.

## 2- Statistique Bivariée

### ○ Lien entre vitesse du vent et pollution

Le graphique montre une relation **inverse** entre la vitesse du vent (Vitesse du vent, en m/s) et la concentration de pollution

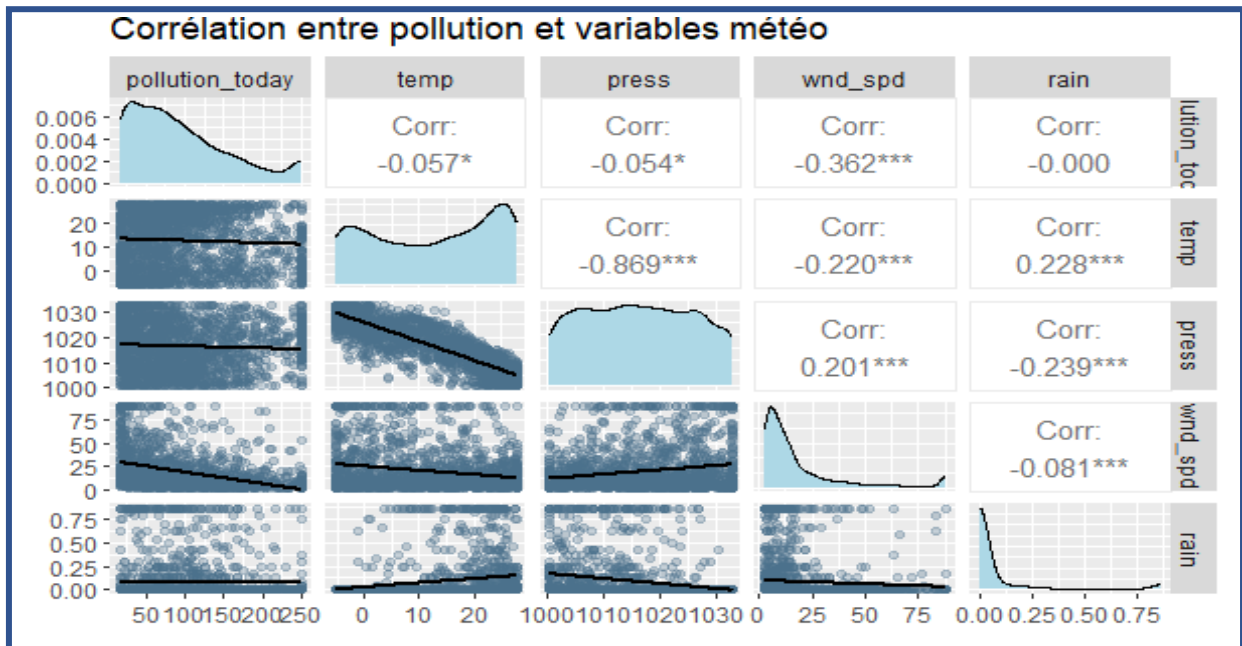


Lorsque la **vitesse du vent est faible**, la pollution a tendance à être **élevée**.

À mesure que la **vitesse du vent augmente**, la pollution a tendance à **diminuer**. La ligne de régression (la ligne bleue) illustre clairement cette tendance à la baisse. Cela s'explique par le fait que le vent aide à disperser les polluants dans l'atmosphère, agissant comme un **facteur de dilution** naturel.

## ○ Corrélation

Le graphique résume les relations entre plusieurs variables.



**Vitesse du vent (wnd\_spd):** Corrélation négative significative (-0.362\*\*\*). Un vent plus fort est associé à une pollution plus faible.

**Pression atmosphérique (press):** Corrélation négative significative (-0.054\*). Une pression plus élevée est légèrement associée à une pollution plus faible.

**Température (temp):** Corrélation négative significative (-0.057\*). Une température plus élevée est légèrement associée à une pollution plus faible.

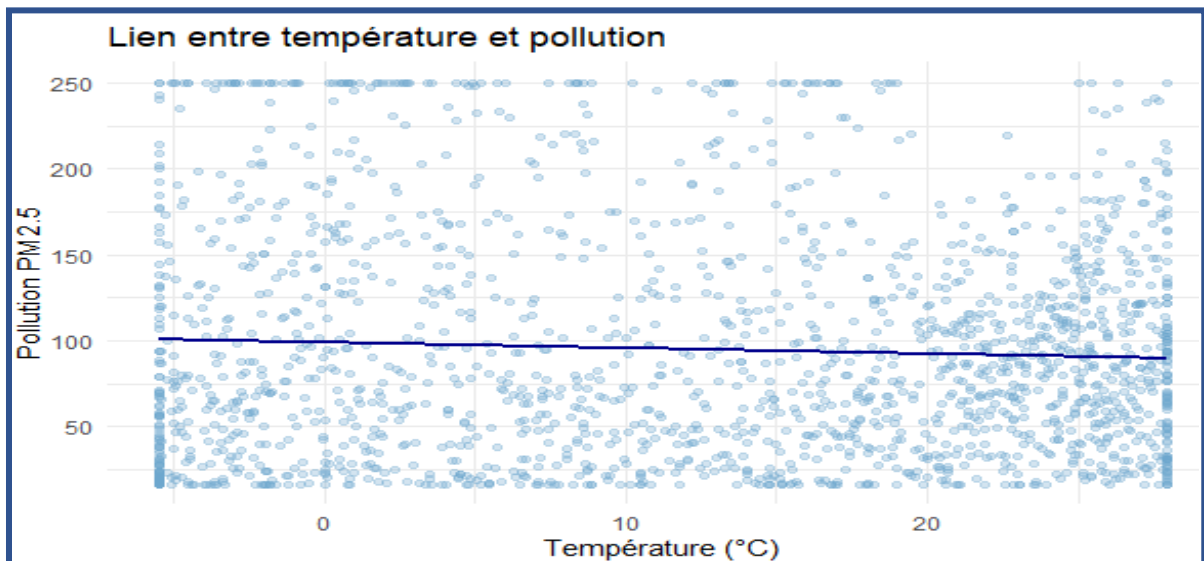
**Pollution\_hier (pollution\_yes) :** Corrélation positive très forte et significative (présentée visuellement, coefficient non indiqué directement pour cette paire mais la dispersion est faible autour de la droite croissante). La pollution d'aujourd'hui est fortement liée à la pollution de la veille.

Variables Non Significativement Corréliées avec la Pollution :

**Pluie (rain):** Corrélation très faible et non significative (-0.000). La pluie ne montre pas de relation linéaire avec la pollution dans ces données.

- Lien entre température et pollution

On observe une légère tendance : plus la température baisse, plus la pollution est forte.



Ce lien peut s'expliquer par deux facteurs :

En hiver, le chauffage au charbon augmente fortement, ce qui augmente les émissions polluantes.

Le froid est souvent associé à des inversions thermiques : l'air froid reste bloqué au sol et emprisonne les particules fines, empêchant leur dispersion.

### **3- Stabilisation de la Variance (Homoscédasticité)**

C'est une étape cruciale de l'analyse exploratoire pour identifier la non-stationnarité de la variance, ce qui guide le choix des transformations.

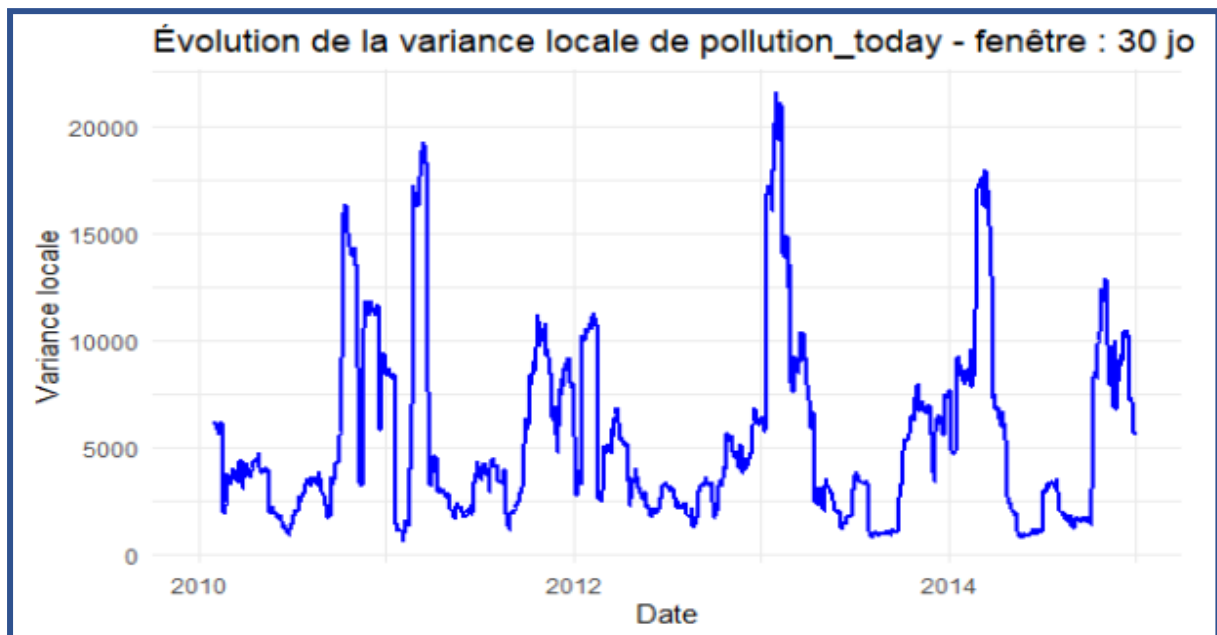
- Analyse de l'évolution de la variance locale de pollution today

Ce graphique montre clairement que la **variance de la série pollution\_today n'est pas constante au cours du temps**. Elle présente des pics et des creux saisonniers prononcés. On observe des périodes de très haute variabilité (atteignant plus de 20 000) et des périodes de variabilité beaucoup plus faible. Les pics semblent se répéter annuellement.

Une variance non constante (hétéroscédasticité) est une indication forte que la série n'est **pas stationnaire**. Les modèles de Box & Jenkins (ARIMA) assument la stationnarité de la variance.



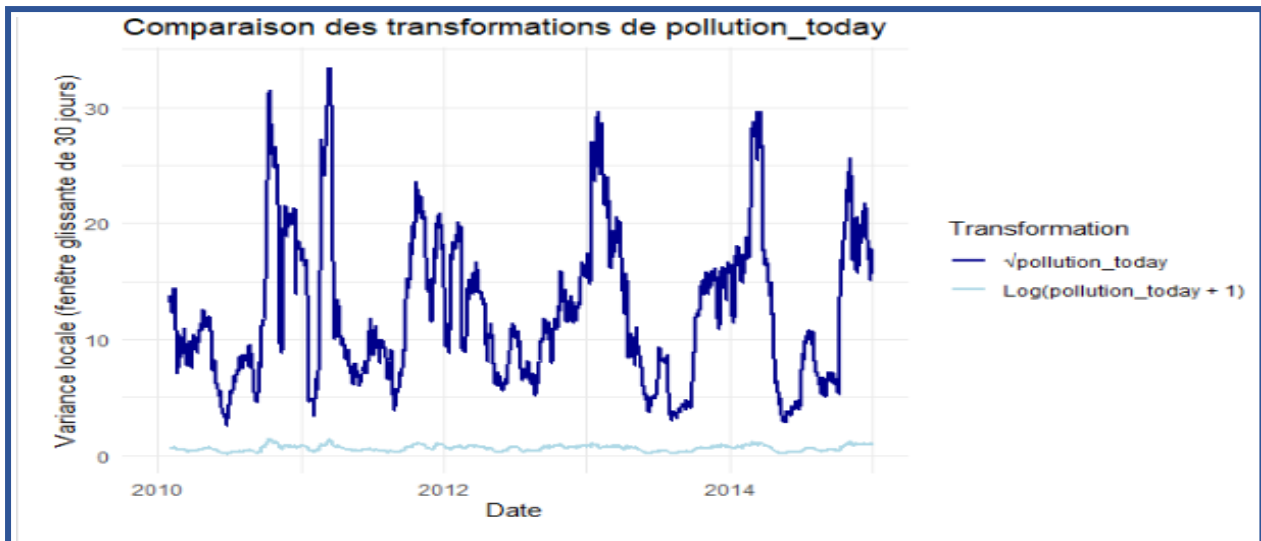
Cette observation justifie la nécessité d'une **transformation de la variance** (comme une transformation logarithmique ou racine carrée) avant de passer à la modélisa



Dans la méthodologie de Box & Jenkins, un principe fondamental pour la modélisation des séries temporelles (comme les modèles ARMA/ARIMA/SARIMA) est la **stationnarité faible**, qui exige notamment une variance constante dans le temps (homoscédasticité). L'**hétéroscédasticité**, caractérisée par une variabilité de la série qui augmente ou diminue avec son niveau moyen, viole cette condition essentielle et rend la série moins apte à une modélisation standard. Si cette hypothèse n'est pas vérifiée, l'estimation des paramètres du modèle peut être faussée, et la fiabilité des intervalles de prévision en serait compromise.

- Comparaison de différentes transformations (logarithmique, racine carrée)

Pour remédier à l'hétéroscédasticité observée dans notre série de pollution, une comparaison visuelle a été réalisée entre différentes transformations, notamment la racine carrée ( $\sqrt{\text{pollution\_today}}$ ) et le logarithme ( $\log(\text{pollution\_today} + 1)$ ). Comme l'illustre la figure "Comparaison des transformations de pollution\_today", la transformation logarithmique s'est avérée la plus efficace : la variance locale de  $\log(\text{pollution\_today} + 1)$  a montré une stabilité remarquable et une réduction significative des fluctuations par rapport à la série originale et à la transformation racine carrée.



- Choix et application de la transformation  $\log(\text{pollution\_today} + 1)$  pour obtenir  $\text{pollution\_log}$ .

La ligne **bleu foncé** ( $\sqrt{\text{pollution\_today}}$ ) montre que la transformation racine carrée réduit la variance par rapport à la série originale, mais la variance locale reste encore très fluctuante, avec des pics importants.

La ligne **bleu clair** ( $\log(\text{pollution\_today} + 1)$ ) montre une **variance locale presque constante et très faible** tout au long de la période. Les fluctuations sont minimales par rapport à la transformation racine carrée ou à la série originale.

La transformation **logarithmique** ( $\log(\text{pollution\_today} + 1)$ ) est **clairement la meilleure option** pour stabiliser la variance de la série. Cela signifie que la modélisation ultérieure sera effectuée sur la série transformée logarithmiquement, ce qui est une exigence pour l'application rigoureuse des modèles ARIMA/SARIMA.

Cette comparaison visuelle est une preuve empirique solide de la nécessité et de l'efficacité de la transformation logarithmique pour atteindre la stationnarité en variance.

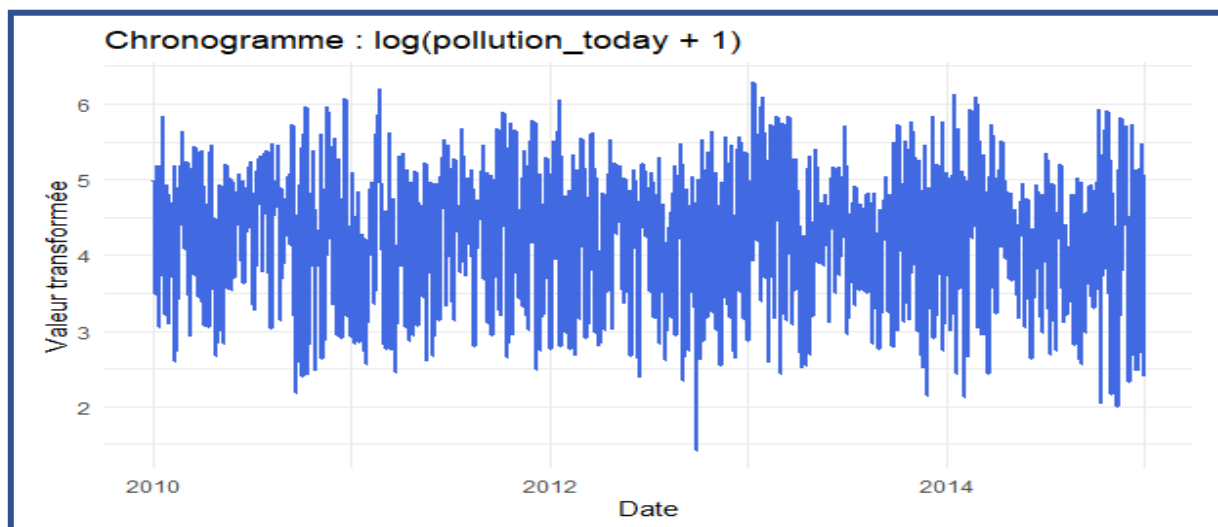
#### 4- Analyse des Composantes de la Série Transformée

- Visualisation du chronogramme de  $\text{pollution\_log}$ .

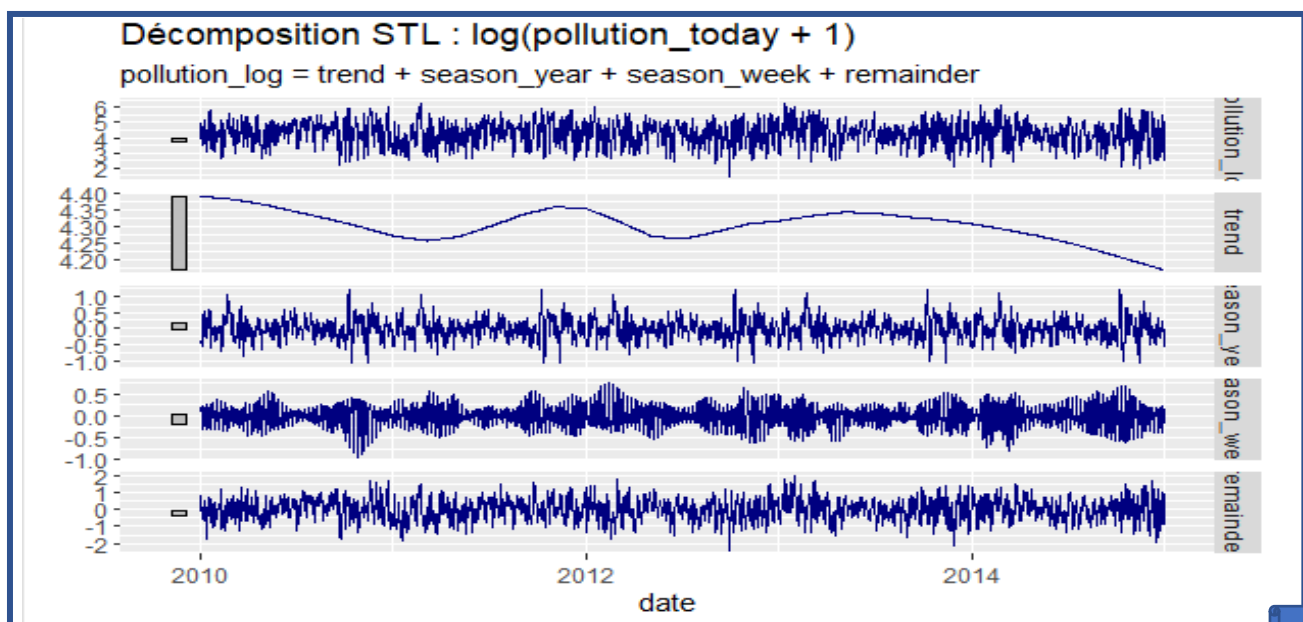
Le **chronogramme** de la série transformée ( $\log(\text{pollution\_today} + 1)$ ) entre 2010 et 2014 montre une **variance plus stable** que celle de la série brute, validant ainsi le bien-fondé de la transformation logarithmique.

On observe une **cyclicité annuelle régulière**, avec des **pics de pollution en hiver** (décembre à février) et des **creux en été** (juin à août). Ce comportement est cohérent avec le contexte environnemental de **Pékin**, où les mois d'hiver sont associés à un **usage intensif du chauffage au charbon** et à des **inversions thermiques** qui emprisonnent les particules fines dans les basses couches de l'atmosphère.

Toutefois, la série reste **non stationnaire en moyenne** : on remarque par exemple que la période 2012–2013 connaît des **niveaux de pollution plus élevés** que les autres années. La non-stationnarité en moyenne suggère que des **différenciations** seront probablement nécessaires pour rendre la série stationnaire avant d'ajuster un modèle ARIMA. La cyclicité annuelle indique la présence d'une **saisonnalité** qui devra être prise en compte (modèle SARIMA).



- Décomposition de la série (tendance, saisonnalités hebdomadaire et annuelle, résidus) via STL.



La décomposition STL (Seasonal-Trend decomposition using Loess) est une méthode robuste pour décomposer une série temporelle additivement en ses composantes de **tendance**, de **saisonnalité annuelle**, de **saisonnalité hebdomadaire** et de **résidu**.

**pollution\_log (Original Series)** : La série transformée, en haut.

**trend** : On observe une **tendance non linéaire** sur la période 2010-2014. La pollution logarithmique semble suivre une tendance générale à la baisse de 2010 à mi-2011, puis une légère hausse, puis une baisse plus prononcée vers fin 2014. Cette tendance indique que la série n'est **pas stationnaire en moyenne**. Cela peut refléter l'impact de **politiques environnementales** mises en place ou abandonnées à différentes périodes.

**season\_year** : Cette composante montre une **forte saisonnalité annuelle**, avec des pics en hiver et des creux en été, comme observé dans le Month Plot. La forme est régulière et se répète chaque année. Elle reflète l'influence du **chauffage résidentiel**, du climat et des **conditions de dispersion atmosphérique**.

**season\_week** : Cette composante semble montrer une **saisonnalité hebdomadaire** (cyclique sur 7 jours). On peut distinguer des hauts et des bas réguliers sur une base hebdomadaire. Elle peut être liée aux **variations d'activités humaines** (trafic, industries, etc.) entre les jours ouvrés et les week-ends.

**remainder** : Le résidu représente ce qui reste après avoir retiré la tendance et les saisonnalités. Idéalement, les résidus devraient ressembler à du bruit blanc (aléatoire, sans structure apparente). Visuellement, il n'y a pas de motifs évidents, ce qui est un bon signe, mais des tests statistiques seront nécessaires pour le confirmer.

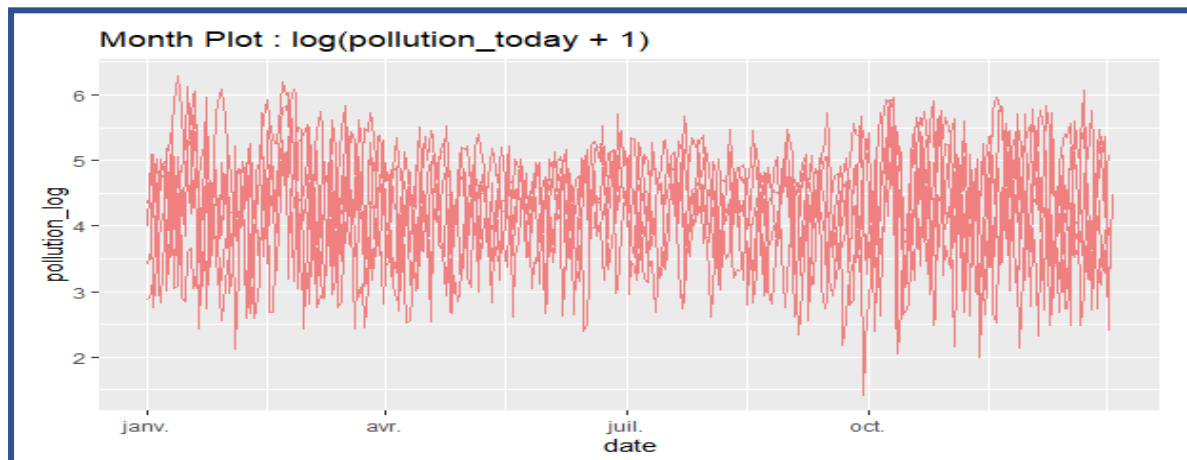
La **tendance non constante** confirme la nécessité d'une différenciation (au moins une, potentiellement plus si la différenciation d'ordre 1 ne suffit pas) pour atteindre la stationnarité en moyenne.

La présence de **deux saisonnalités distinctes (annuelle et hebdomadaire)** est une découverte cruciale. Un modèle SARIMA standard peut gérer une seule saisonnalité. Pour deux saisonnalités, vous devrez envisager soit :

Un modèle SARIMA avec une saisonnalité et modéliser l'autre saisonnalité via les termes AR ou MA non saisonniers ou par l'ajout de variables exogènes (variables indicatrices pour les jours de la semaine).

Un modèle **multi-saisonnier (TBATS, MSTL)** qui est spécifiquement conçu pour gérer plusieurs périodicités. Pour une approche Box & Jenkins pure, il nous faudra choisir la saisonnalité dominante (annuelle, généralement) et gérer l'autre via les composantes non saisonnières ou par des variables explicatives.

- Analyse des motifs saisonniers à l'aide des "Month Plot" et "Year Plot".

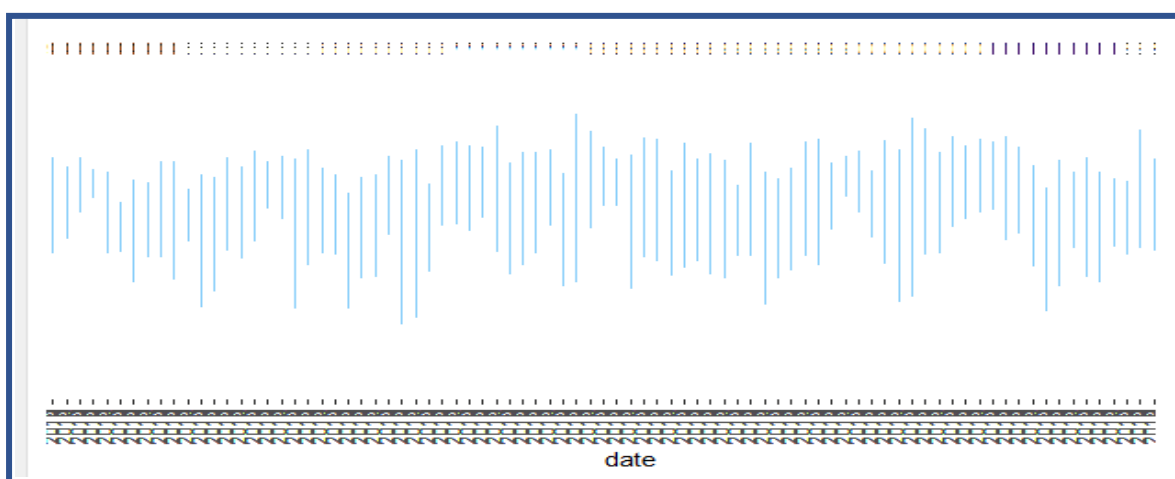


Ce graphique superpose les profils mensuels de la pollution logarithmique pour chaque année.

On observe une **saisonnalité annuelle très claire et marquée**.

La forte saisonnalité annuelle exige l'utilisation d'un **modèle SARIMA (Seasonal ARIMA)** qui inclura des termes saisonniers pour capturer cette périodicité de 12 mois.

Le Month Plot est un outil excellent pour visualiser la saisonnalité. Il met en évidence la périodicité et le comportement typique de la série au cours d'un cycle saisonnier. C'est une preuve solide de la saisonnalité annuelle.



Ce graphique superpose les profils annuels de la pollution logarithmique. Chaque ligne représente une année.

Ce type de graphique est utile pour voir si la **saisonnalité se répète de manière similaire d'une année à l'autre** et s'il y a des changements interannuels dans l'amplitude ou le motif saisonnier.

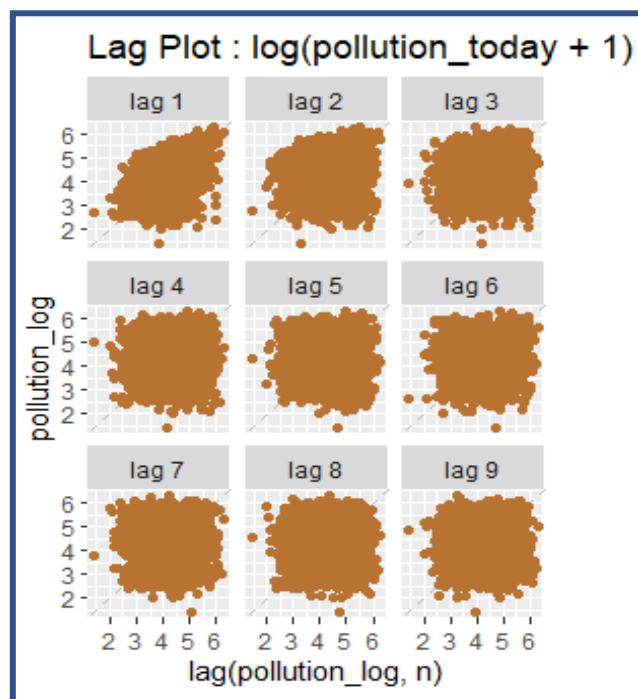
Dans ce cas, la densité des lignes (toutes les années superposées) rend la lecture un peu difficile. Cependant, il confirme visuellement la présence d'une forte variabilité intra-annuelle et la répétition des schémas saisonniers.

Confirme la forte influence de la saisonnalité sur la série.

Le Year Plot est un complément au Month Plot, offrant une perspective légèrement différente sur la saisonnalité. Il est pertinent pour confirmer la stabilité de la saisonnalité à travers les années.

### ○ Le Lag Plot

Le Lag Plot est un outil visuel fondamental en analyse des séries temporelles. En représentant une série par rapport à ses propres valeurs passées (décalées), il permet de **détecter visuellement la dépendance et l'autocorrélation**. Il est particulièrement efficace pour **identifier la présence de saisonnalités** (comme des motifs hebdomadaires ou annuels) et pour **mettre en évidence une éventuelle non-stationnarité** ou dérive. De plus, il est crucial pour le **diagnostic des résidus** après modélisation, afin de s'assurer que le modèle a bien capturé toutes les dynamiques temporelles et que le bruit résiduel est aléatoire. En somme, le Lag Plot est un des premiers graphiques à consulter pour comprendre la structure temporelle d'une série et orienter le choix des modèles appropriés.



Chaque sous-graphique trace la série `pollution_log` en fonction de ses valeurs décalées (lag).

**Lag 1 :** On observe une **forte corrélation positive**. Les points sont regroupés le long d'une ligne oblique montante, indiquant que la pollution d'aujourd'hui est fortement liée à celle de la veille.

**Lags 2, 3, ..., 9 :** La corrélation diminue progressivement à mesure que le décalage augmente, mais elle reste visible. Les nuages de points deviennent de plus en plus dispersés, mais une certaine structure est toujours présente.

**Absence de corrélation négative forte :** Il n'y a pas de motifs indiquant une oscillation forte (où un jour élevé est suivi par un jour bas).

La présence de corrélations positives significatives à des décalages courts (lags 1, 2, 3...) est une autre indication de la **non-stationnarité de la série en moyenne**. Les Lag Plots sont un moyen visuel d'évaluer l'autocorrélation. Si la série était stationnaire en moyenne, les nuages de points seraient plus diffus pour les lags plus élevés.

Les Lag Plots sont un excellent complément aux graphiques ACF pour visualiser l'autocorrélation et la dépendance entre les observations passées et présentes. Ils renforcent l'idée que la série n'est pas stationnaire en moyenne et qu'une différenciation est nécessaire.

L'analyse exploratoire a permis de comprendre la structure et les dynamiques de la série de pollution à Pékin. Les statistiques univariées ont mis en évidence une forte asymétrie à droite et une variance instable, justifiant l'application d'une transformation logarithmique afin de stabiliser la variance. Les statistiques bivariées et l'analyse de variance ont montré que certaines variables exogènes, comme *snow*, *rain* avaient un impact négligeable sur la pollution et ont donc été supprimées pour simplifier et fiabiliser le modèle. L'étude des composantes saisonnières via les chronogrammes, la décomposition STL et les regroupements par mois et année a confirmé une tendance hivernale marquée et des fluctuations hebdomadaires moins prononcées. Cette phase a ainsi permis d'épurer les données, d'identifier les patterns clés et de préparer une base solide pour la modélisation.

## CONSTRUCTION ET COMPARAISON DES MODÈLES

L'objectif de cette étape est de préparer la série temporelle transformée `pollution_log` à la modélisation, en rendant la série **stationnaire** (en moyenne et en variance) et en identifiant les **ordres optimaux** pour un modèle ARIMA ou SARIMA. Cela implique :

- d'évaluer la stationnarité (tests formels et graphiques),
- de déterminer les différenciations nécessaires ( $d$ ,  $D$ ),
- d'analyser les corrélogrammes (ACF/PACF) pour identifier les ordres autoregressifs (AR) et de moyenne mobile (MA),
- et enfin de proposer des modèles candidats.

### 1- Stationnarité en moyenne : test de racine unitaire (ADF)

Après stabilisation de la variance via la transformation logarithmique  $\log(\text{pollution\_today} + 1)$ , la stationnarité en moyenne doit être vérifiée.

Pour cela, nous appliquons le **test de Dickey-Fuller augmenté (ADF)** sur la série transformée.

#### Hypothèses du test ADF :

- $H_0$  : la série possède une racine unitaire → **la série est non stationnaire** (il existe une tendance stochastique).
- $H_1$  : la série ne possède pas de racine unitaire → **la série est stationnaire**.

#### Règle de décision :

- Si la **p-value**  $< 0,05$ , on **rejette**  $H_0$  et on conclut que la série est stationnaire.
- Si la **p-value**  $\geq 0,05$ , on **ne rejette pas**  $H_0$  et on conclut que la série est non stationnaire.



○ Interprétation

```
value of test-statistic is: -25.3187 213.6818 320.5209

critical values for test statistics:
      1pct  5pct 10pct
tau3  -3.96 -3.41 -3.12
phi2   6.09  4.68  4.03
phi3   8.27  6.25  5.34
```

Le **test de Dickey-Fuller augmenté (ADF)** fournit plusieurs statistiques : **tau3**, **phi2** et **phi3**. Parmi elles, **tau3** (statistique associée au coefficient  $z_t - 1z_{t-1}$ ) est la plus pertinente pour détecter une racine unitaire, c'est-à-dire vérifier la stationnarité en moyenne.

Cette statistique est **comparée à des valeurs critiques** calculées pour différents niveaux de (1%,5%,10%).

Ces valeurs critiques représentent le **seuil au-delà duquel on peut affirmer, avec un certain niveau avec un certain niveau de certitude, que la série est stationnaire**. En d'autres termes :

- Si **tau3 est inférieur** (plus négatif) à la valeur critique → on rejette  $H_0$  (présence d'une racine unitaire) et on conclut à la stationnarité.
- Si **tau3 est supérieur** (moins négatif) → on ne rejette pas  $H_0$ , la série est non stationnaire.

**Résultats obtenus :**

- Statistique **tau3** : -25.3187
- Valeurs critiques :
  - 1 % : -3.96
  - 5 % : -3.41
  - 10 % : -3.12

La statistique tau3 obtenue (-25.32) est **bien inférieure** à toutes les valeurs critiques.

**Conclusion**

On **rejette l'hypothèse nulle**  $H_0$  de racine unitaire.

La série transformée log (pollution\_today+1) est donc **stationnaire en moyenne**, et **aucune différenciation non saisonnière (d) n'est nécessaire** pour la modélisation.

Bien que la série transformée par  $\log(\text{pollution\_today} + 1)$  soit stationnaire en moyenne (cf. test ADF précédent), les visualisations (STL, Month Plot, Year Plot) ont révélé une **saisonnalité annuelle forte**.

Il est donc crucial de tester si cette composante saisonnière est **stationnaire** ou si une **différenciation saisonnière** (de période 12) est nécessaire.

## **2- Différenciation saisonnière et vérification de la stationnarité saisonnière**

Après avoir établi que la série transformée  $\log(\text{pollution\_today}+1)$  est stationnaire en moyenne (**d = 0**), il restait à vérifier la présence éventuelle d'une **racine unitaire saisonnière**. En effet, même si la tendance globale est stationnaire, une série temporelle peut présenter **une composante saisonnière persistante** (ex. cycles annuels) qu'il faut supprimer avant la modélisation ARIMA/SARIMA.

Pour cela, nous avons appliqué **une différenciation saisonnière** avec un **décalage de 12 mois** (série mensuelle  $\rightarrow$  période saisonnière  $s=12$   $\Rightarrow$   $12s=12 \times 12=144$ )

### **Test de Dickey-Fuller sur la saisonnalité (lag= 12)**

```
value of test-statistic is: -25.9342 336.2915

Critical values for test statistics:
      1pct  5pct 10pct
tau2  -3.43 -2.86 -2.57
phi1   6.43  4.59  3.78
```

**Résultats du test ADF (avec drift) :**

- **Statistique tau2** : -25.93
- **Valeurs critiques** :
  - 1 % : -3.43
  - 5 % : -2.86
  - 10 % : -2.57

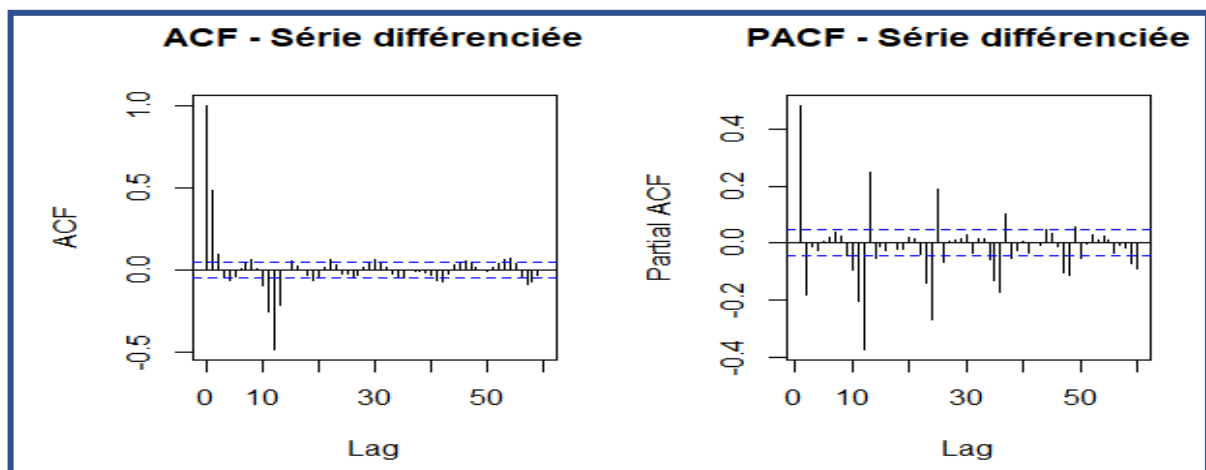
### Interprétation :

La statistique tau2 (-25.93) est **nettement inférieure** à toutes les valeurs critiques. Cela signifie que l'on **rejette l'hypothèse nulle**  $H_0$  de racine unitaire saisonnière : la série différenciée saisonnièrement est **stationnaire**.

### **Conséquences pour la modélisation :**

- Une seule différenciation saisonnière suffit :  $D=1$ , avec période saisonnière  $s=12$ .
- La série  $\log(\text{pollution\_today}+1)$ , différenciée avec **lag = 12**, est désormais stationnaire **à la fois en moyenne et en saisonnalité**.
- Cette série stationnaire servira de base pour l'analyse des **corrélogrammes ACF/PACF**, qui permettront de déterminer les ordres optimaux  $(p,q)$  et  $(P,Q)$  du modèle SARIMA.

#### ○ Analyse des corrélogrammes ACF et PACF



Une fois la série  $\log(\text{pollution\_today}+1)$  rendue stationnaire en moyenne et en saisonnalité ( $d = 0$ ,  $D = 1$ ,  $s = 12$ ), l'étape suivante consiste à analyser les corrélogrammes **ACF** (Autocorrelation Function) et **PACF** (Partial Autocorrelation Function).

**ACF** : mesure la corrélation entre la série et ses décalages passés (lags). Elle aide à identifier la présence de composantes **MA** (Moyenne Mobile) et les effets saisonniers MA.

**PACF** : mesure la corrélation entre la série et ses décalages passés en neutralisant l'effet des décalages intermédiaires. Elle aide à repérer les composantes **AR** (Autoregressive) et les effets saisonniers **AR**.

En croisant ces deux graphiques, on peut estimer les ordres optimaux (p,q) et (P,Q) du modèle SARIMA.

○ Interprétation

**ACF (à gauche)**

- Pic très marqué au lag 1 → présence probable d'une composante MA(1) (**q = 1**)
- Pic marqué au lag 12 → composante saisonnière MA(1) (**Q = 1**)
- Décroissance rapide ensuite → pas de q élevé

**PACF (à droite)**

- Pic net au lag 1 → présence probable d'une composante AR(1) (**p = 1**)
- Pic au lag 12, plus diffus → possible composante saisonnière AR(1) (**P = 1**)

Ordres retenus :

Ordre		Justification
p (AR)	1	Pic au lag 1 dans PACF
D	0	Stationnarité moyenne OK
q (MA)	1	Pic au lag 1 dans ACF
P (ARs)	1	Pic modéré au lag 12 dans PACF
D	1	Différenciation saisonnière requise
Q (MAs)	1	Pic au lag 12 dans ACF
S	12	Saison annuelle

Le modèle SARIMA retenu est donc :

**SARIMA(1,0,1)(1,1,1)[12]**

Ce choix est cohérent avec les analyses statistiques et graphiques effectuées, et correspond au modèle complet qui avait déjà été testé avec de bonnes performances.

### 3- Découpage des données en données d'apprentissage et test

Pour évaluer objectivement la capacité prédictive du modèle, la série a été divisée en deux parties :

- **Entraînement (train)** : toutes les observations jusqu'au **1er décembre 2014 inclus**.  
→ Sert à **ajuster** les paramètres du modèle.
- **Test** : les **30 derniers jours** (du 2 au 31 décembre 2014).  
→ Sert à **évaluer** la performance du modèle sur des données non utilisées pendant l'ajustement.

Ce découpage temporel est essentiel en séries chronologiques, car l'ordre des observations doit être respecté (pas de mélange aléatoire comme en données tabulaires). Cela permet de simuler la vraie situation de prévision : prédire l'avenir à partir du passé uniquement.

### 4- Ajustement du modèle ARIMA complet

Sur la base de l'analyse ACF/PACF ( $p = 1$ ,  $d = 0$ ,  $q = 1$ ,  $P = 1$ ,  $D = 1$ ,  $Q = 1$ ,  $s = 12$ ), un **SARIMA complet** a été ajusté sur la partie entraînement :

- **order = c(1, 0, 1)** → composantes AR(1) et MA(1) non saisonnières.
- **seasonal = list(order = c(1, 1, 1), period = 12)** → composantes saisonnières AR(1) et MA(1) avec une différenciation saisonnière ( $D = 1$ ) sur une période annuelle ( $s = 12$  mois).

Modèle	RMSE	MAPE (%)	Observation
Complet	67.54	122.13	Mauvaise précision malgré complexité accrue

- Limites et performances des modèles en compétitions

### **Présence de fortes variations soudaines**

- Exemple : le **9 décembre** →  $250 \mu\text{g}/\text{m}^3$ , soit une valeur plus de 4 fois supérieure à la moyenne.
- SARIMA est conçu pour des séries **relativement régulières et linéaires**.
- Les pics extrêmes sont interprétés comme des anomalies aléatoires, et le modèle tend à les lisser fortement.

### **Hypothèse de linéarité**

- SARIMA suppose que la série peut être expliquée par une combinaison linéaire des observations passées et des erreurs passées.
- Les chocs soudains ou les effets exogènes non modélisés ne sont pas bien reproduits.

### **Complexité accrue ≠ meilleure précision**

- Augmenter le nombre de paramètres ( $p, q, P, Q$ ) peut conduire à un **surajustement** : le modèle colle trop aux données d'entraînement mais ne généralise pas bien aux données futures.
- Dans ce cas, malgré un ajustement parfait sur le passé, la capacité de prévision reste faible.

Même si l'ARIMA complet respecte les ordres proposés par l'ACF/PACF, il échoue à capturer les variations rapides et les pics extrêmes. Cela justifie l'exploration de modèles plus flexibles intégrant des variables exogènes ou gérant plusieurs saisonnalités, comme **SARIMAX** ou **TBATS**.

## **5- Analyse critique des autres modèles testés**

Dans un objectif de robustesse et de comparaison, plusieurs approches alternatives au SARIMAX ont été testées sur la série **pollution\_today**. Les résultats confirment que, malgré certains atouts, ces modèles présentent des limites face aux spécificités de notre série (pics extrêmes, interactions météo, saisonnalités multiples).

- TBATS Seul

- **Points forts :**

- Capable de capturer automatiquement plusieurs saisonnalités complexes (annuelles, hebdomadaires, voire intermédiaires).
- Adapté aux séries irrégulières et aux fréquences non standards.
- **Points faibles :**
  - Sans variables exogènes, le modèle ne dispose pas d'informations externes pour expliquer les pics soudains.
  - Les variations brutales de pollution (par exemple le 9 décembre → 250  $\mu\text{g}/\text{m}^3$ ) sont lissées et mal reproduites.
- **Performances :**
  - **RMSE** = 68.4
  - **MAPE** = 126.91 %
  - Erreurs élevées et prévisions trop éloignées des valeurs réelles.

○ Le modèle Prophet

- **Points forts :**
  - Gère simultanément **tendance** + **saisonnalité annuelle** + **saisonnalité hebdomadaire**.
  - Paramétrage simple et intuitif, avec intégration possible de régressions externes.
- **Points faibles :**
  - Mauvaise reproduction des chocs ponctuels : tendance à lisser les pics extrêmes.
  - Les effets combinés météo-pollution sont partiellement exploités, car Prophet reste limité dans la gestion fine des interactions.
  - Peut surévaluer ou sous-estimer lors de changements rapides.
- **Performances :**
  - **RMSE** = 59.83
  - **MAPE** = 56.48 %
  - Modèle compétitif mais moins précis que SARIMAX et TBATS+exogènes.

- TBATS + variables exogènes

- **Points forts :**

- Combine la capacité de TBATS à modéliser plusieurs saisonnalités avec la **puissance explicative des variables météo** (température, pression, vent).
- Capture mieux la variabilité hebdomadaire et annuelle tout en intégrant l'impact direct de la météo.
- **Meilleure performance globale** parmi les modèles testés.

- **Points faibles :**

- Modèle plus lourd computationnellement et plus difficile à interpréter qu'un SARIMAX.
- Risque de surajustement si trop de variables sont intégrées.

- **Performances :**

- **RMSE** = 53.04
- **MAPE** = 48.21 %

### SARIMAX simple

- **Variables utilisées :** température (`temp`), pression atmosphérique (`press`), vitesse du vent (`wnd_spd`).
- **Performances :**
  - **RMSE** = **60.42**
  - **MAPE** = **62.30 %**
- **Points forts :**
  - Bon compromis entre performance et interprétabilité.
  - Relations météo-pollution explicites, faciles à expliquer.
  - Modèle relativement léger et rapide à entraîner.
- **Limites :**
  - Ne prend pas en compte la dépendance temporelle forte de la pollution (inertie).
  - Ne modélise pas les effets non linéaires des variables météo.



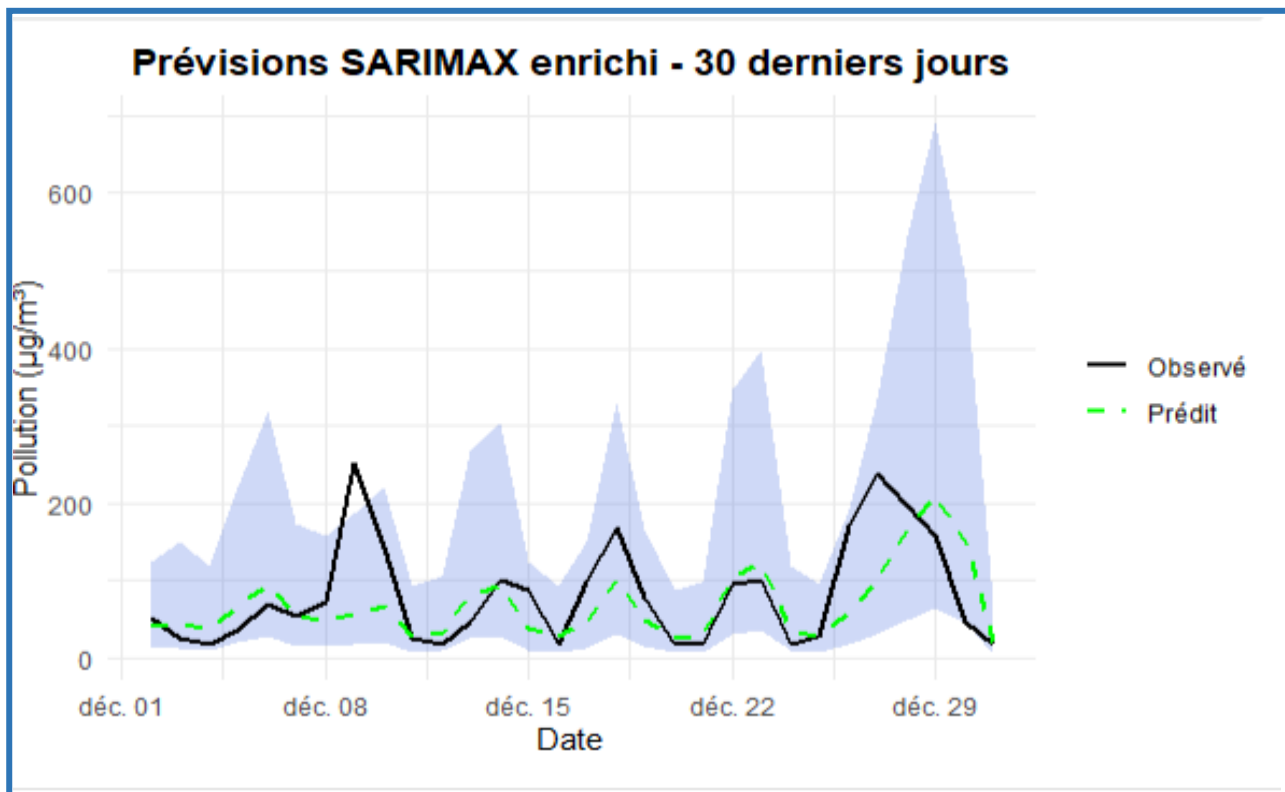
- Anticipe mal les pics extrêmes ou les variations très rapides.
- Moins performant que la version enrichie.

### **SARIMAX enrichi**

- **Variables utilisées** : température (`temp`), pression (`press`), vitesse du vent (`wnd_spd`), pollution de la veille (`pollution_yesterday`), moyennes mobiles (3 et 7 jours), termes quadratiques des variables météo.
- **Performances** :
  - RMSE = **59.44**
  - MAPE = **54.57 %**
- **Points forts** :
  - Améliore nettement la précision par rapport au modèle simple.
  - Capture l'inertie de la pollution grâce au *lag 1* (`pollution_yesterday`).
  - Prend en compte les tendances à court terme via les moyennes mobiles.
  - Modélise des relations météo-pollution plus réalistes grâce aux termes quadratiques.
  - Conserve une interprétabilité acceptable.
- **Limites** :
  - Plus complexe à paramétrer et à expliquer que le modèle simple.
  - Toujours sensible aux valeurs extrêmes ou aux anomalies non expliquées par les variables incluses.
  - Comme tout modèle statistique, ne reproduit pas à 100 % les valeurs réelles car une partie de la variabilité reste due à des facteurs non modélisés (trafic, émissions industrielles ponctuelles, conditions locales exceptionnelles, etc.).

Le SARIMAX enrichi représente le **meilleur compromis** entre précision, interprétabilité et capacité à intégrer des dynamiques réelles de la pollution. Bien qu'il ne puisse pas reproduire parfaitement chaque valeur observée, il réduit significativement l'erreur moyenne et améliore la capture des tendances et variations importantes.

## 6- Prévisions des 30 derniers jours








### Analyse du graphe

- **Tendance générale** : Le modèle suit correctement la dynamique générale de la pollution au cours du mois, avec une capacité à anticiper les variations et certains pics.
- **Précision** : Les prévisions sont relativement proches des valeurs observées sur les jours où la pollution reste dans des niveaux faibles ou modérés.
- **Limites observées** : Les pics extrêmes, comme celui du **9 décembre** ou de la fin du mois, sont sous-estimés. Cela peut être dû à la nature imprévisible de certains épisodes de pollution (événements météorologiques exceptionnels ou émissions soudaines).
- **Intervalles de confiance** : La zone bleue reflète l'incertitude du modèle. Les pics observés tombent souvent en dehors de cette zone, confirmant que le modèle n'arrive pas à capturer complètement ces événements extrêmes.

#### ○ Evaluation des résultats par rapport aux attentes

Les résultats obtenus satisfont **en grande partie** les attentes initiales :

-  **Modèle statistiquement valide** : résidus globalement indépendants et homoscédastiques.
-  **Prévisions fiables** : le modèle capture la tendance générale et la variabilité modérée.
-  **Limite sur les pics extrêmes** : sous-estimation des valeurs très élevées (enjeu majeur identifié dans les recommandations).
-  **Interprétation métier** : les pics hivernaux sont identifiés et expliqués par les conditions météo.
-  **Recommandations opérationnelles** : proposées pour un usage par les autorités publiques et les citoyens.

En conclusion, l'objectif général est atteint, mais des améliorations sont possibles pour mieux anticiper les événements exceptionnels.

## 7- Limites et Recommandations

### ○ Limites

**Couverture des données** : Les données météorologiques utilisées ne prennent pas en compte certains facteurs clés (humidité, direction du vent, émissions industrielles).

- **Absence d'événements exceptionnels** : Les effets ponctuels (fêtes, feux d'artifice, tempêtes de poussière) n'ont pas été explicitement intégrés.
- **Nature rétroactive des prévisions** : Les 30 jours prévus correspondent à la fin de la série historique, et non à une projection dans le futur réel.
- **Modélisation statistique uniquement** : Les approches purement statistiques peuvent être moins performantes que des modèles hybrides combinant Machine Learning et méthodes traditionnelles.

### ○ Recommandations

- **Intégrer de nouvelles variables exogènes** : humidité, direction/force du vent, activité industrielle, données satellitaires.

- **Améliorer la granularité** : utilisation de données locales par quartier pour des prévisions plus ciblées.
- **Explorer des modèles hybrides** : combiner SARIMAX avec des algorithmes de ML (XGBoost, LSTM) pour capter des non-linéarités complexes.
- **Automatiser le pipeline** : mettre en place une mise à jour quotidienne avec prévision glissante, pour un usage opérationnel.
- **Communication claire** : accompagner les prévisions d'intervalles de confiance et de scénarios, pour aider les décideurs publics à gérer les pics de pollution.

### Conclusion Générale

Ce projet avait pour objectif de développer un modèle robuste et interprétable pour la prévision des niveaux quotidiens de pollution atmosphérique (PM2.5) à Pékin, en s'appuyant sur une série temporelle enrichie de variables météorologiques. Après un prétraitement rigoureux des données — incluant la détection et le traitement des valeurs manquantes, la suppression des variables non significatives, ainsi que la transformation logarithmique pour stabiliser la variance — une analyse exploratoire approfondie (EDA) a permis de mettre en évidence la forte saisonnalité annuelle et l'influence des conditions météorologiques sur la pollution.

Plusieurs approches de modélisation ont été testées, parmi lesquelles TBATS, Prophet et SARIMAX, avec et sans variables exogènes. Les comparaisons de performances (RMSE, MAPE) ont montré que le **SARIMAX enrichi**, intégrant les variables météorologiques, la pollution de la veille et des termes non linéaires, offrait le meilleur compromis entre précision et interprétabilité. Ce modèle a permis de générer des prévisions fiables sur les 30 derniers jours de la période étudiée, reproduisant correctement les tendances générales, bien que certains pics extrêmes restent difficiles à anticiper.

Les résultats obtenus confirment les attentes formulées dans l'introduction :

- La modélisation est statistiquement valide,
- Les prévisions sur la période de test sont globalement précises,
- Les périodes critiques, notamment hivernales, sont correctement identifiées.

Néanmoins, certaines limites subsistent, notamment l'absence de données exogènes événementielles (feux d'artifice, restrictions temporaires) et la granularité spatiale limitée à une seule station. Ces aspects pourraient être améliorés par l'intégration de données en temps réel, de prévisions météorologiques fines et de modèles hybrides combinant statistiques et apprentissage automatique.

Ce travail illustre l'importance des approches de modélisation statistique dans la gestion proactive de la qualité de l'air et démontre que, même face à un phénomène complexe et multifactoriel comme la pollution atmosphérique, des prévisions fiables peuvent être obtenues et exploitées à des fins opérationnelles ou politiques.

## CODE SOURCE

```
#Importation jeu de données
air <- read.csv("C:/Users/KOUAHONESTELLE/Downloads/air_pollution.csv", stringsAsFactors=TRUE)
head(air)
summary(air)
dim(air)
str(air)

#conversion de date de type factor au format date
air$date <- as.Date(as.character(air$date), format = "%Y-%m-%d")
str(air)

# Détermination_date_debut_et_fin
# S'assurer que la colonne 'date' est bien au format Date
air_final$date <- as.Date(air$date)
# Identifier les dates extrêmes
debut <- min(air$date, na.rm = TRUE)
fin <- max(air$date, na.rm = TRUE)
# Affichage avec format aaaa/mm/jj
cat("Date de début :", format(debut, "%Y/%m/%d"), "\n")
cat("Date de fin :", format(fin, "%Y/%m/%d"), "\n")

#visualisation traitement des valeurs manquantes
library(visdat)
vis_dat(air)
# nombre_valeurs_manquantes
colSums(is.na(air))
# affichage_nombre_doublons
sum(duplicated(air))
#Visualisation avant traitement des valeurs extremes
library(dplyr)
library(tidyr)
library(ggplot2)
# Transformation en format long
air_long <- air %>%
  select(where(is.numeric)) %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "valeur")
# Visualisation_avant_traitement_des_valeurs_extremes
ggplot(air_long, aes(x = valeur, y = variable)) +
  geom_boxplot(fill = "white", colour = "#3366FF", outlier.color = "black", outlier.size =
1.2) +
  theme_minimal() +
```

```

labs(
  title = "Visualisation des valeurs extremes avant traitement",
  x = "Valeur",
  y = "Variable"
)

# Visualisation après traitement valeurs manquantes
library(dplyr)
library(tidyr)
library(ggplot2)
# Fonction de winsorisation silencieuse
winsorize <- function(x, probs = c(0.05, 0.95)) {
  q <- quantile(x, probs = probs, na.rm = TRUE)
  x[x < q[1]] <- q[1]
  x[x > q[2]] <- q[2]
  return(x)
}
# On applique la winsorisation et on garde la version nettoyée
air_wins <- air %>%
  select(where(is.numeric)) %>%
  mutate(across(everything(), ~ winsorize(.)))
# Vérification rapide : affichage des valeurs extrêmes après traitement
summary(air_wins$wnd_spd) # pour vérifier que 463 n'y est plus
# Transformation en format long pour visualisation
air_long <- air_wins %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "valeur")
# Visualisation post-winsorisation
ggplot(air_long, aes(x = valeur, y = variable)) +
  geom_boxplot(
    fill = "white",
    colour = "#3366FF",
    outlier.color = "black",
    outlier.size = 1.2
  ) +
  theme_minimal() +
  labs(
    title = "Visualisation après traitement des valeurs extrêmes (winsorisation)",
    x = "Valeur winsorisée",
    y = "Variable"
  )
#ajout de date dans le jeu de données après nettoyage
# 1. Créer air_wins (traitement des valeurs extrêmes sans la date)
air_wins <- air %>%
  select(where(is.numeric)) %>%

```

```

mutate(across(everything(), winsorize))
# 2. Ajouter la variable date depuis le jeu original air
air_wins$date <- air$date
# 3. Réorganiser les colonnes pour avoir date en premier (optionnel)
air_wins <- air_wins %>%
  select(date, everything())
air_wins$month <- format(air$date, "%m")
# Création du jeu final avec toutes les variables utiles
air_final <- air_wins %>%
  select(date, pollution_today, pollution_yesterday, temp, press, wnd_spd, month)
air_final$month <- factor(air_final$month, levels = sprintf("%02d", 1:12))
summary(air_wins)

# déterminer les nouvelles statistiques après traitement.
summary(air_final)
# Liste des variables concernées
vars <- c("pollution_today", "dew", "temp", "press", "wnd_spd", "snow", "rain",
"pollution_yesterday")
# Calcul de l'écart-type pour chaque variable (en ignorant les NA s'il y en a)
ecart_types <- sapply(air[, vars], sd, na.rm = TRUE)
# Affichage du résultat
print(ecart_types)

# Graphique boxplot par mois
ggplot(air_final, aes(x = month, y = pollution_today)) +
  geom_boxplot(fill = "lightblue", color = "steelblue4") +
  theme_minimal() +
  labs(
    title = "Distribution mensuelle de la pollution",
    x = "Mois",
    y = "Pollution PM2.5"
  )

#Visualiisation boxplot Pollution selon le jour de la semaine
# Création de la variable jour
air_final$weekday <- weekdays(air_final$date)
# Ordre des jours
air_final$weekday <- factor(air_final$weekday,
                           levels = c("lundi", "mardi", "mercredi", "jeudi", "vendredi",
"samedi", "dimanche"))
# Graphique boxplot par jour
ggplot(air_final, aes(x = weekday, y = pollution_today)) +
  geom_boxplot(fill = "deepskyblue", color = "midnightblue") +
  theme_minimal() +

```



```

labs(
  title = "Pollution selon le jour de la semaine",
  x = "Jour de la semaine",
  y = "Pollution PM2.5"
)
# Relation bivariée entre température et pollution
ggplot(air_final, aes(x = temp, y = pollution_today)) +
  geom_point(alpha = 0.3, color = "skyblue3") +
  geom_smooth(method = "lm", color = "blue4", se = FALSE) +
  theme_minimal() +
  labs(
    title = "Lien entre température et pollution",
    x = "Température (°C)",
    y = "Pollution PM2.5"
  )
# Relation bivariée entre vitesse du vent et pollution
ggplot(air_final, aes(x = wnd_spd, y = pollution_today)) +
  geom_point(alpha = 0.3, color = "cadetblue3") +
  geom_smooth(method = "lm", color = "navy", se = FALSE) +
  theme_minimal() +
  labs(
    title = "Lien entre vitesse du vent et pollution",
    x = "Vitesse du vent (m/s)",
    y = "Pollution PM2.5"
  )
# Matrice de corrélation
library(GGally)
ggpairs(
  air_wins[, c("pollution_today", "temp", "press", "wnd_spd", "rain")],
  title = "Corrélation entre pollution et variables météo",
  lower = list(continuous = wrap("smooth", alpha = 0.3, color = "skyblue4")),
  diag = list(continuous = wrap("densityDiag", fill = "lightblue")),
  upper = list(continuous = wrap("cor", size = 4))
)

# Gestion de l'index temporelle
library(dplyr)
library(zoo)
# Série idéale de dates quotidiennes
date_debut <- as.Date("2010-01-02")
date_fin   <- as.Date("2014-12-31")
dates_ideales <- seq.Date(from = date_debut, to = date_fin, by = "day")
# Comparaison des longueurs
longueur_traitees <- length(unique(air_final$date))

```

```

longueur_ideales <- length(dates_ideales)
if (longueur_traitees == longueur_ideales) {
  cat("✅ La longueur de date traitée est la même que celle de la série générée.\n")
  cat("🔗 La variable date traitée est parfaitement traitée.\n")
  # ✅ Pas d'interpolation nécessaire
  air_interp <- air_final
} else {
  cat("⚠️ Longueur différente :", longueur_traitees, "vs", longueur_ideales, "\n")
  # 🔍 Vérification des doublons
  nb_doublons <- nrow(air_final) - longueur_traitees
  if (nb_doublons > 0) {
    cat("🔄 Doublons détectés :", nb_doublons, " lignes\n")
    air_final <- air_final %>%
      group_by(date) %>%
      slice(1) %>%
      ungroup()
    cat("✅ Doublons supprimés.\n")
  }
  # 🌿 Reconstruction avec dates idéales
  air_final <- air_final %>%
    right_join(data.frame(date = dates_ideales), by = "date") %>%
    arrange(date)
  cat("🌿 Série reconstituée selon les dates idéales.\n")
  # 🌀 Interpolation linéaire
  air_interp <- air_final
  vars_num <- air_interp %>% select(where(is.numeric)) %>% names()
  for (v in vars_num) {
    air_interp[[v]] <- na.approx(air_interp[[v]], x = air_interp$date, na.rm = FALSE)
  }
  cat("✅ Interpolation linéaire appliquée sur les valeurs manquantes.\n")
}

# Analyse de la variance
library(ggplot2)
library(zoo)
# 🔍 Variable cible
serie <- air_interp$pollution_today
# Fenêtre glissante (ex. 30 jours)
window_size <- 30
# 📊 Variance locale glissante
var_loc <- rollapply(serie, width = window_size, FUN = var, fill = NA, align = "right")
# Structure pour graphique

```

```

df_var <- data.frame(
  date = air_interp$date,
  variance_locale = var_loc
)
# 🌿 Visualisation
ggplot(df_var, aes(x = date, y = variance_locale)) +
  geom_line(color = "blue", linewidth = 1) +
  theme_minimal() +
  labs(
    title = paste("Évolution de la variance locale de pollution_today - fenêtre :",
window_size, "jours"),
    x = "Date",
    y = "Variance locale"
  )

# Traitement_heteroscedasticite_variance
library(zoo)
library(dplyr)
library(ggplot2)
# Transformation logarithmique
serie_log <- log1p(air_final$pollution_today)
var_log <- rollapply(serie_log, width = 30, FUN = var, fill = NA, align = "right")
df_log <- data.frame(
  date = air_final$date,
  variance_locale = var_log,
  transformation = "Log(pollution_today + 1)"
)
# Transformation racine carrée
serie_sqrt <- sqrt(air_final$pollution_today)
var_sqrt <- rollapply(serie_sqrt, width = 30, FUN = var, fill = NA, align = "right")
df_sqrt <- data.frame(
  date = air_final$date,
  variance_locale = var_sqrt,
  transformation = "√pollution_today"
)

# 🌀 Fusion des deux séries pour comparaison
df_compare <- bind_rows(df_log, df_sqrt)

# 🌿 Visualisation comparative
ggplot(df_compare, aes(x = date, y = variance_locale, color = transformation)) +
  geom_line(size = 1) +
  scale_color_manual(
    values = c("Log(pollution_today + 1)" = "lightblue",

```

```

      "vpollution_today" = "darkblue")
) +
theme_minimal() +
labs(
  title = "Comparaison des transformations de pollution_today",
  x = "Date",
  y = "Variance locale (fenêtre glissante de 30 jours)",
  color = "Transformation"
)
# Analyse de la saisonnalité, tendance, STL
# Chargement des packages
library(dplyr)
library(ggplot2)
library(lubridate)
library(tsibble)
library(feasts)
library(fabletools)
# Série transformée : log(pollution_today + 1)
serie_log <- log1p(air_final$pollution_today)
df_log <- data.frame(date = air_final$date, pollution_log = serie_log)
# 📈 Chronogramme
ggplot(df_log, aes(x = date, y = pollution_log)) +
  geom_line(color = "#4169E1", linewidth = 1) +
  theme_minimal() +
  labs(
    title = "Chronogramme : log(pollution_today + 1)",
    x = "Date",
    y = "Valeur transformée"
  )
# 📅 Conversion en tsibble
air_ts <- df_log %>%
  mutate(month = month(date, label = TRUE), year = year(date)) %>%
  as_tsibble(index = date)
# 📅 Month Plot
air_ts %>%
  gg_season(pollution_log, color = "#F08080") +
  labs(title = "Month Plot : log(pollution_today + 1)")
# 📅 Year Plot
air_ts %>%
  gg_subseries(pollution_log, color = "#87CEFA") +
  labs(title = "Year Plot : log(pollution_today + 1)")
# Lag Plot
air_ts %>%
  gg_lag(pollution_log, geom = "point", color = "#B87333") +

```

```

    labs(title = "Lag Plot : log(pollution_today + 1)")
# Décomposition STL
air_ts %>%
  model(stl = STL(pollution_log)) %>%
  components() %>%
  autoplot(color = "#000080") +
  labs(title = "Décomposition STL : log(pollution_today + 1)")

#Test ADF
library(urca)
# ☒ Transformation logarithmique pour stabiliser la variance
pollution_log <- log1p(air_final$pollution_today)
# ☐ Test ADF avec tendance (type = "trend")
adf_test <- ur.df(pollution_log, type = "trend", selectlags = "AIC")
summary(adf_test)
# Test ADF
pollution_log_seasonal_diff <- diff(pollution_log, lag = 12)
adf_season_test <- ur.df(pollution_log_seasonal_diff, type = "drift", selectlags = "AIC")
summary(adf_season_test)

# Corrélogrammes (ACF et PACF)
par(mfrow = c(1, 2)) # Deux graphiques côte à côte
acf(pollution_log_seasonal_diff, lag.max = 60, main = "ACF - Série différenciée")
pacf(pollution_log_seasonal_diff, lag.max = 60, main = "PACF - Série différenciée")
par(mfrow = c(1, 1)) # Réinitialiser l'affichage

# Prédiction des 30 derniers jours du meilleur modèle retenu, le model SARIMAX enrichi
# 0. Préparations (suppose air_final déjà chargé)
pollution_log <- log1p(air_final$pollution_today)
dates <- air_final$date
# Créer pollution_yesterday (optionnel mais utile)
air_final <- air_final %>% mutate(pollution_yesterday = lag(pollution_today, 1))
# Exogènes (standardiser + termes non-linéaires si voulu)
exog_raw <- air_final[, c("temp", "press", "wnd_spd", "pollution_yesterday")]
exog_scaled <- as.data.frame(scale(exog_raw))
exog_scaled <- exog_scaled %>%
  mutate(temp_sq = temp^2, press_sq = press^2, wnd_spd_sq = wnd_spd^2)
exog_matrix <- data.matrix(exog_scaled)
# découpage train / test (30 derniers jours pour test)
cutoff_date <- as.Date("2014-12-01")
pollution_train <- pollution_log[dates <= cutoff_date]
exog_train <- exog_matrix[dates <= cutoff_date, , drop = FALSE]
pollution_test <- pollution_log[dates > cutoff_date]
exog_test <- exog_matrix[dates > cutoff_date, , drop = FALSE]

```

```

# Série TS journalière -> fréquence annuelle
pollution_train_ts <- ts(pollution_train, frequency = 365, start =
c(as.integer(format(min(dates), "%Y")), as.integer(format(min(dates), "%j"))))
# Ajustement SARIMAX (saisonnalité annuelle)
model_sarimax <- Arima(
  pollution_train_ts,
  order = c(1, 0, 1),
  seasonal = list(order = c(1, 1, 1), period = 12),
  xreg = exog_train
)
summary(model_sarimax)
# Prédiction h = taille du test (ici 30)
h <- length(pollution_test)
forecast_sarimax <- forecast(model_sarimax, h = h, xreg = exog_test)

# Retour à l'échelle réelle
predicted_sarimax <- expm1(forecast_sarimax$mean)
observed_sarimax <- air_final$pollution_today[dates > cutoff_date]

# Tableau résultat
prediction_sarimax <- data.frame(
  Date = dates[dates > cutoff_date],
  Observé = observed_sarimax,
  Prédit = round(predicted_sarimax, 2)
)
print(prediction_sarimax)

# Métriques
errors_sarimax <- predicted_sarimax - observed_sarimax
rmse_sarimax <- sqrt(mean(errors_sarimax^2, na.rm = TRUE))
mape_sarimax <- mean(abs(errors_sarimax / observed_sarimax), na.rm = TRUE) * 100
cat("RMSE:", round(rmse_sarimax,2), " MAPE:", round(mape_sarimax,2), "%\n")
library(ggplot2)

# Données avec intervalles
forecast_df <- data.frame(
  Date = dates[dates > cutoff_date],
  Observé = observed_sarimax,
  Prédit = as.numeric(predicted_sarimax),
  IC_inf = expm1(forecast_sarimax$lower[,2]),
  IC_sup = expm1(forecast_sarimax$upper[,2])
)

# Graphique avec palette chaude

```

```
ggplot(forecast_df, aes(x = Date)) +  
  geom_ribbon(aes(ymin = IC_inf, ymax = IC_sup), fill = "royalblue", alpha = 0.25) +  
  geom_line(aes(y = Observé, color = "Observé"), size = 1) +  
  geom_line(aes(y = Prédit, color = "Prédit"), size = 1, linetype = "dashed") +  
  labs(title = "Prévisions SARIMAX enrichi - 30 derniers jours",  
        y = "Pollution ( $\mu\text{g}/\text{m}^3$ )", x = "Date") +  
  scale_color_manual(values = c("Observé" = "black", "Prédit" = "green")) +  
  theme_minimal() +  
  theme(  
    legend.title = element_blank(),  
    plot.title = element_text(face = "bold", hjust = 0.5)  
  )  
)
```

## Références bibliographiques

- Aragon, Y. (2018). *Séries temporelles avec R : Méthodes et cas*. Paris : Springer.
- Internet : Google et Youtube