

Rapport de stage de fin d'études
Sujet : Détection d'anomalies et maintenance prédictive



Par : Aminata Ndiaye
Entreprise : Groupe Renault
Maître de stage : Vincent Feuillard

Remerciements

Je remercie l'ensemble de l'équipe data science et IA pour m'avoir accueillie chaleureusement et m'avoir offert l'opportunité de travailler au sein de leur équipe. Leur partage de connaissances et leur expertise tout au long de mon stage ont été inestimables.

Je tiens à exprimer ma reconnaissance envers Vincent Feuillard, pour avoir supervisé ce stage et avoir permis son déroulement en toute fluidité. Je le remercie pour sa pédagogie, son aide et son accompagnement tout au long de mon stage.

Pour finir, je souhaite remercier l'ensemble du corps professoral de l'université Paris Dauphine pour leurs enseignements.

Table des matières

Remerciements	1
Introduction	3
1 Présentation de l'entreprise	4
1.1 Le Technocentre Renault	4
1.2 DQSC - Direction Qualité Satisfaction Client - Equipe Data Science & IA	4
2 Détection de dérives sur les demandes de remboursements garantis	5
2.1 Présentation du sujet	5
2.2 Les Scan statistics sur variables discrètes	6
2.2.1 Idée générale	6
2.2.2 Formalisation	7
2.2.3 Construction du test et implémentation	7
2.2.4 Tailles de fenêtre multiples	8
2.2.5 Illustrations sur données simulées	12
2.3 Scan statistics sur variables continues	15
2.3.1 Idée Générale	15
2.3.2 Formalisation	15
2.3.3 Illustrations sur données simulées	18
3 Maintenance prédictive des Presses d'emboutissage des usines	21
3.1 Présentation du sujet	21
3.2 Description des données	21
3.3 Analyse des données et approche envisagée	22
3.4 Construction d'indicateurs pertinents	22
3.4.1 Point de rupture et modèle de rupture dans la tendance	23
3.4.2 L'algorithme PELT	24
3.4.3 Application	25
Conclusion	27

Introduction

Dans le cadre de mon cursus à l'université Paris Dauphine, j'ai pu effectuer mon stage de fin d'études au sein de la direction qualité satisfaction client d'un acteur majeur de l'industrie automobile en France et dans le monde, le Groupe Renault.

L'industrie automobile, en constante évolution et d'une complexité croissante, repose de plus en plus sur l'exploitation efficace des données pour améliorer la qualité, la sécurité et la performance des véhicules. Parallèlement, l'intelligence artificielle connaît un engouement important au sein de la communauté scientifique, notamment dans le secteur de l'automobile. Mon stage a été une opportunité exceptionnelle de plonger au cœur de ce secteur innovant et d'observer l'importance stratégique de l'analyse statistique dans le processus de conception, de fabrication et de gestion de la qualité des véhicules.

L'objectif principal était de mettre en pratique mes connaissances en statistique pour analyser des données réelles et fournir des informations précieuses aux autres métiers de la direction afin de faciliter la prise de décision. Ce rapport vise à documenter en détail mon parcours au sein de l'entreprise, en mettant en évidence les projets sur lesquels j'ai travaillé, les méthodes statistiques utilisées et les résultats obtenus.

Après une très brève présentation de l'entreprise, je vais exposer dans ce rapport les différents aspects de mon stage, en commençant par une description de ma mission au sein du département de la qualité, suivi de la description de mes travaux liés aux problématiques de manufacturing.

1 Présentation de l'entreprise

1.1 Le Technocentre Renault

Le Technocentre, situé à Guyancourt en région parisienne, est le premier centre de recherche et de développement automobile en Europe. Il rassemble depuis 1998 tous les acteurs impliqués dans la conception et le développement des véhicules de l'Alliance et du Groupe Renault, soit à ce jour 12 000 personnes. C'est au Technocentre que sont définies les gammes des marques Renault, Dacia et Renault Korea Motors (auparavant Renault Samsung Motors). Installé sur 150 hectares de terrain, le Technocentre est la plus grande entreprise de la communauté d'agglomération de Saint-Quentin-en-Yvelines et le 1er centre de R&D en France.

Depuis sa création en 1998, le Technocentre est au coeur du développement des futurs véhicules du Groupe Renault. C'est le centre de la conception, de la haute technicité et de l'innovation de l'entreprise.

1.2 DQSC - Direction Qualité Satisfaction Client - Equipe Data Science & IA

L'équipe Data Science & IA de la DQSC, la Direction de Qualité et Satisfaction Client, se consacre entre autres à l'analyse des incidents sur les véhicules dans le but d'améliorer la qualité des produits. Elle joue également un rôle crucial dans le suivi de la satisfaction client tout au long de la vie des véhicules après leur livraison.

L'équipe que j'ai intégrée est principalement constituée de statisticiens et de data scientists et a pour mission fondamentale de fournir un soutien statistique essentiel pour le suivi des produits commercialisés par le groupe Renault. Son objectif principal consiste à mettre en place des outils statistiques pour surveiller la qualité des produits et la satisfaction client.

2 Détection de dérives sur les demandes de remboursements garantis

2.1 Présentation du sujet

Dans le cadre du suivi de la qualité des véhicules du groupe Renault, nous analysons ce que nous appelons des DRG : demandes de remboursements garantis. Il s'agit d'un ensemble d'informations renseigné par les garagistes lorsque les véhicules encore sous garantie constructeur retournent chez le concessionnaire à cause de défaillances. Des informations diverses s'y trouvent, telles que la date de l'ordre de réparation, le VIN (numéro d'identification du véhicule), les commentaires des experts métiers, le nombre de mois de mise en service, etc.

A chaque DRG sera associée un NITG (numéro d'incident technique garanti), il s'agit d'un référentiel utilisé pour désigner un élément ou une partie du véhicule. C'est sur ce référentiel que nous chercherons à détecter des anomalies. Ainsi, chaque DRG sera associée à un NITG responsable à la défaillance.

NITG	VIN	km	Mois en service	Marque	...	Date d'ouverture d'OR
XX1	VF09876KI097863	8765	11	REN	...	13-01-2022
...
XX1	VF05417600986763	3885	3	DAC	...	25-10-2022
...
XX2	VF134576LM0989753	765	3	REN	...	02-01-2023
...
XX1	VA2P436N090778	20705	15	REN	...	17-12-2022
...
XXX	VL966KD981219	3765	5	DAC	...	27-11-2022

TABLE 1 – Exemple de table des DRG. Chaque ligne représente une DRG, la première colonne indique le NITG associé à la DRG.

Pour chaque NITG, nous observons la distribution des DRG (ou incidents) associées aux NITG en fonction du temps. Pour établir la distribution des DRG par NITG, il est nécessaire de spécifier la date de référence de l'étude. Il peut s'agir de la date de construction du véhicule ou encore de la date d'ouverture de l'ordre de réparation.

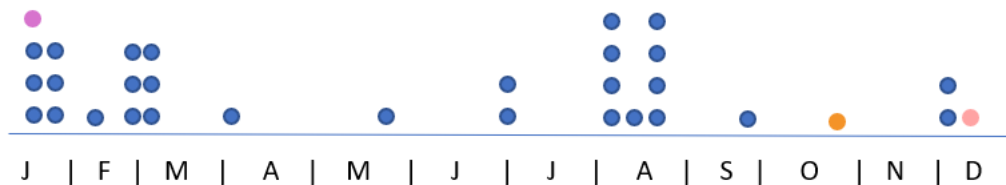


FIGURE 1 – Exemple de distribution des DRG du NITG XX1 au cours de l'année 2022 par date d'ouverture de l'ordre de réparation. Les points colorés correspondent aux lignes de la même couleur dans la table 1.

Notre objectif est de d'identifier de possibles dérives dans la distribution du nombre d'incidents. Nous nous intéresserons plus particulièrement au cas où le nombre d'incidents est supérieur à la normale. Nous allons procéder à une analyse rétrospective des DRG pour déterminer les NITG présentant un nombre trop important de défaillances (les NITG suspects).

Pour mesurer cette "normalité", nous utiliserons les scan statistics. Dans la partie suivante, nous présenterons cet outil et sa formalisation mathématique. Nous montrerons par la suite comment ce dernier nous aide dans la modélisation et dans la résolution de notre problème.

Dans le cadre de mon stage, j'ai principalement travaillé sur l'extension de l'outil de scan statistics pour des tailles de fenêtre multiples et sur le développement d'un algorithme de scan statistics pour des données continues.

L'outil de scan statistic pour une fenêtre fixe et des données discrètes avait déjà été implémenté. J'ai pu l'appliquer sur un historique de données de façon à qualifier sa pertinence avec l'aide d'experts en traitement des incidents (incidentologues). J'ai aussi contribué à l'automatisation des alertes pour qu'elles soient exploitables en temps continu par l'ensemble des incidentologues (mise à jour hebdomadaire).

2.2 Les Scan statistics sur variables discrètes

2.2.1 Idée générale

Une "scan statistic" ou "window statistic" permet de détecter un cluster de points dans un espace. Dans le cadre de notre étude, nous allons considérer le cas univarié temporel.

Le premier écrit sur les scan statistics date de 1963 avec "Clustering of random points in line and plane" publié par Naus [1]. Il est considéré comme le père de cette discipline.

Dans ses premières applications, les scan statistics ont été utilisées en épidémiologie, en santé publique et en astronomie. Aujourd'hui, elles s'étendent à la biologie moléculaire, aux sciences sociales, et dans notre cas, le contrôle qualité.

Nous considérons une période $[0, T]$ avec $T \in \mathbb{R}$, N points ou évènements sont distribués sur cette période. Nous introduisons :

- ω , une fenêtre temporelle telle que $\omega \in [0, T]$.
- S_ω , le plus grand nombre d'évènements dans une fenêtre de taille ω .
- W_k , la plus petite période contenant k points (ou évènements).

S_ω est le cluster maximal que nous appelons également scan statistic.

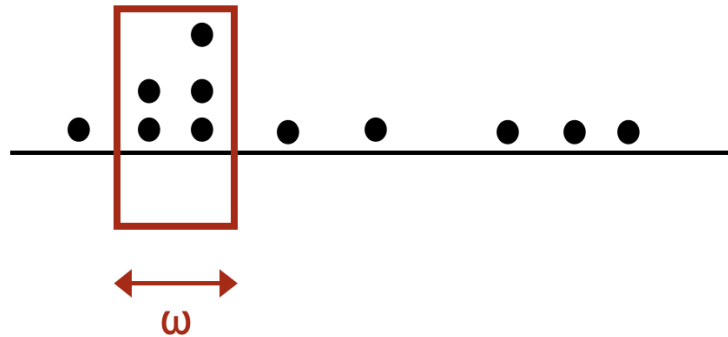


FIGURE 2 – Représentation d'une distribution d'évènements. Le rectangle rouge représente l'emplacement de la fenêtre ω tel que le nombre d'éléments soit maximal. Ainsi $S_\omega = 5$.

Pour déterminer si le cluster est anormal, nous devons construire une statistique de test à partir des hypothèses suivantes :

- H_0 : le cluster maximal est typique pour une distribution uniforme des évènements, il n'y a pas d'anomalie.
- H_a : le cluster maximal est atypique pour une distribution uniforme des évènements, il y a une anomalie.

2.2.2 Formalisation

Nous considérons l'espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$, avec $\Omega = \mathbb{R}$ et $\mathcal{F} = \mathcal{B}(\mathbb{R})$ la tribu borélienne.

Soit $N \in \mathbb{N}$, le nombre d'évènements qui est supposé connu, et $X = (X_1, \dots, X_N) \in \mathbb{R}^N$ la variable aléatoire de dimension N telle que, pour tout $i \in [1, N]$, X_i représente la localisation temporelle de l'évènement i . De plus, $X_i \perp\!\!\!\perp X_j$ pour tout $(i, j) \in [1, N]^2$ tel que $i \neq j$.

Sous l'hypothèse nulle, les X_i sont distribuées uniformément sur l'intervalle $[0, T]$. Nous cherchons à calculer $\mathbb{P}(S_\omega \geq k)$ qui est la probabilité d'observer un cluster de taille k ou plus sachant qu'il y a N évènements. Par la suite nous noterons cette probabilité $\mathbf{P}(k, N, \omega)$

Nous pouvons maintenant établir les relations entre S_ω et W_k

$$\mathbb{P}(S_\omega \geq k) = \mathbb{P}(W_k \leq \omega)$$

Cette probabilité commune est souvent notée de la façon suivante pour montrer sa dépendance à la variable N :

$$\mathbf{P}(k, N, \omega) = 1 - \mathbf{Q}(k, N, \omega)$$

Approximation : Afin d'approximer la probabilité de la scan statistic $\mathbf{P}(k, N, \omega)$ nous utilisons la formule de Wallenstein et Neff (1987) que nous pouvons retrouver à la page 12 du livre de Naus [2].

2.2.3 Construction du test et implémentation

La scan statistic nous permet de mesurer le degré d'atypie d'un cluster. Nous construisons le test statistique de la façon suivante.

Si $\mathbb{P}(S_\omega \geq k)$ est grande (supérieure au niveau α) alors le cluster n'est pas atypique, l'hypothèse nulle n'est pas rejetée. Il n'y a pas d'anomalie dans la distribution d'après le test.

En revanche, si $\mathbb{P}(S_\omega \geq k)$ est inférieure ou égale à α , c'est à dire que la probabilité d'observer un cluster de k éléments ou plus est faible, alors l'hypothèse nulle est rejetée. Il y a une anomalie d'après le test.

Pour l'implémentation de l'algorithme, nous introduisons les paramètres suivants :

- *kind_of_date* : La date de référence temporelle de l'étude.
- *N_last_day* : Le nombre N de jours à prendre en compte pour constituer le dataset.
- *Scanning_window* : La taille de la fenêtre ω
- *min_n_tot* : Le nombre minimale d'évènements (incidents ou DRG) pour considérer qu'un cluster "suspect"(atypique) doit être signalé.

- *alpha_scan* : Nombre de mois précédant la date d'étude pendant lequel si l'analyse identifie un cluster, ce dernier est signalé. En dehors de cette période l'anomalie est considérée comme trop ancienne. Il n'y a pas de signalement.

L'étude est réalisée à la date T .

Data: Table des DRG, liste des NITG (*list_nitg*), paramètres

Result: Liste de NITG suspects

Initialiser la liste des NITG suspects;

for *nitg* **in** *list_nitg* **do**

 Calculer S_ω ;

 Calculer P^* ;

if $P^* \leq 0.5$ **and** $S_\omega \geq \min_n_tot$ **and** $W_k \in]T - \alpha_scan \times 30, T]$ **then**

 Ajouter *nitg* à la liste des NITG suspects

end

end

Algorithm 1: Algorithme de détections de NITG suspects basé sur les scan statistics.

2.2.4 Tailles de fenêtre multiples

Jusqu'à présent, la taille de la fenêtre ω était considérée comme fixe. Il s'agissait d'un paramètre de l'algorithme. Contrairement aux autres paramètres, qui ont un sens du point de vu suivi qualité, comme *kind_of_date* qui permet de choisir le type de date de référence pour l'étude, la taille de la fenêtre n'a pas de sens du point de vue suivi qualité et doit être optimisée pour détecter au mieux les dérives.

Le choix de la taille de la fenêtre peut avoir une influence assez importante sur les résultats de l'algorithme. C'est ce qui est illustrée sur les figures 3 et 4.

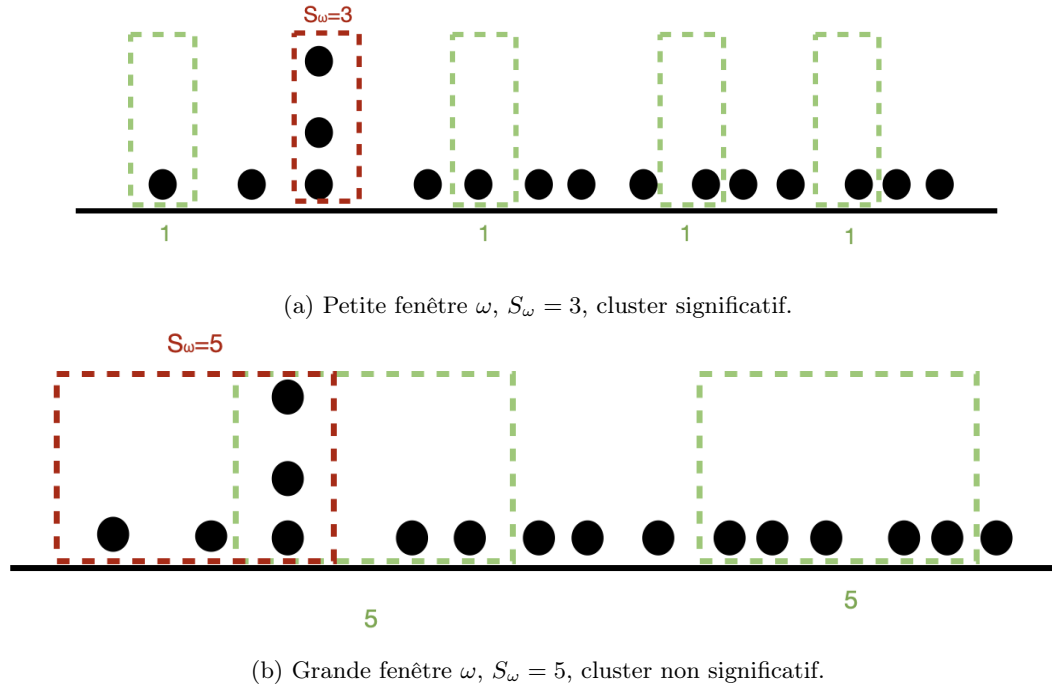
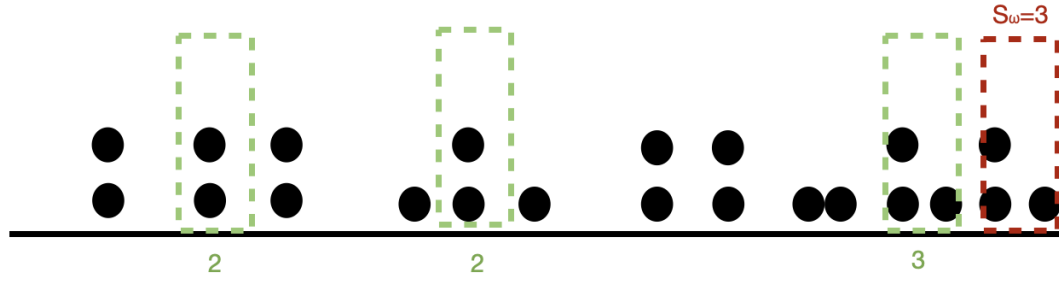
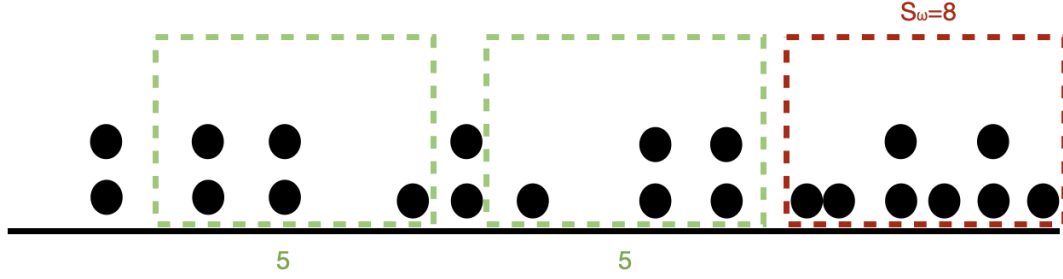


FIGURE 3 – Identification d'un cluster significatif avec petite fenêtre ω mais non significatif avec grande fenêtre ω .



(a) Petite fenêtre ω , $S_{\omega} = 3$, cluster non significatif.



(b) Grande fenêtre ω , $S_{\omega} = 8$, cluster significatif.

FIGURE 4 – Identification d'un cluster significatif avec grande fenêtre ω mais non significatif avec petite fenêtre ω .

Pour palier ce problème, nous introduisons une procédure de tests multiples afin de tester plusieurs tailles de fenêtre. Il est crucial de pouvoir contrôler le taux de fausses détections.

Dans le cas de la taille de fenêtre fixe, le taux de faux positifs était α , en effet le test statistique était de niveau α . Dans le cas de tailles de fenêtre multiples, nous cherchons à contrôler le FWER (family wise error rate) qui correspond à la probabilité de faire au moins une erreur de type I dans la famille de tests. Nous testons m taille de fenêtre, nous procédons donc à m tests. Soit :

- m_0 , le nombre de vraies hypothèses nulles.
- $m - m_0$, le nombre de vraies hypothèses alternatives.
- V , le nombre de faux positif (erreur de première espèce).
- U , le nombre de vrais négatifs.
- S , le nombre de vrais positifs.
- T , le nombre de faux négatifs.

	L'hypothèse nulle est vraie	L'hypothèse nulle est fausse	Total
Rejet de l'hypothèse nulle	V	S	R
Non rejet de l'hypotèse nulle	U	T	$m - R$
Total	m_0	$m - m_0$	m

Le FWER est défini comme la probabilité de faire au moins une erreur de première espèce, cette erreur doit être inférieure à α .

$$FWER = \mathbb{P}(V \geq 1) \leq \alpha$$

Un travail bibliographique a été réalisé sur les algorithmes de scan statistic sur des tailles de fenêtre multiples, mais ces algorithmes n'ont pas été retenus.

La première raison est que certaines procédures permettaient de déterminer si une distribution est suspecte ou non en testant plusieurs tailles de fenêtre, mais ne permettaient pas de retrouver l'emplacement de la fenêtre associée au cluster (voir références [3] [4]). Ce qui est embêtant du point de vue exploitation des résultats. Il est important de pouvoir identifier la période d'anomalie pour identifier la cause et trouver potentiellement une solution du point de vue incidentologie.

La seconde raison pour laquelle ces procédures n'ont pas été retenues est que certaines procédures ne permettaient pas de calculer une approximation de la probabilité de la scan statistic sans passer par une procédure de Monte Carlo, qui est coûteuse et moins exacte du point de vue algorithmique. (voir référence [5])

Nous nous sommes donc tournés vers des procédures de tests multiples corrigés du type Bonferroni.

Afin de contrôler le FWER, deux corrections sont envisagées : Bonferroni et Holm. Les procédures de correction de type I sont utilisées dans le cadre des tests statistiques multiples pour ajuster les niveaux de signification afin de prendre en compte le fait que plus il y a de test qui sont effectués, plus les chances d'obtenir des résultats significatifs par pur hasard augmentent. Cela aide à maintenir un taux global d'erreur de type I au niveau choisi.

Soit H_1, \dots, H_m les hypothèses nulles associées aux m tests et P_1, \dots, P_m les p-valeurs des m tests. Nous notons $P_{(1)}, \dots, P_{(m)}$ les p-valeurs des m tests rangées dans l'ordre croissant et $H_{(1)}, \dots, H_{(m)}$ les hypothèses nulles correspondantes.

1. Procédure de Bonferroni :

- C'est l'une des méthodes les plus simples et les plus conservatrices.
- Elle divise simplement le niveau de signification α par le nombre total de tests effectués. Nous testons :

$$P_i < \frac{\alpha}{m} \quad i = 1, \dots, m.$$

- .
- Cela maintient l'erreur de type I globale à un niveau α , mais peut être trop conservatrice et réduire la puissance des tests.

2. Procédure de Holm :

- Elle ordonne les p-valeurs de manière croissante, puis ajuste les niveaux de signification de manière séquentielle en fonction de cet ordre.

Nous testons :

$$P_{(i)} < \frac{\alpha}{m + 1 - i}$$

Si l'inégalité est vérifiée, nous rejetons $H_{(i)}$, et nous passons au test suivant d'hypothèse nulle $H_{(i+1)}$. Si, au contraire, l'inégalité n'est pas vérifiée, $H_{(i)}$ n'est pas rejetée et toutes les hypothèses nulles des tests suivants également.

- Elle est plus puissante que Bonferroni.

La figure 5 provenant du livre "An Introduction to Statistical Learning with applications in Python" [6] nous montre bien la différence de puissance entre la procédure de Holm et celle de Bonferroni.

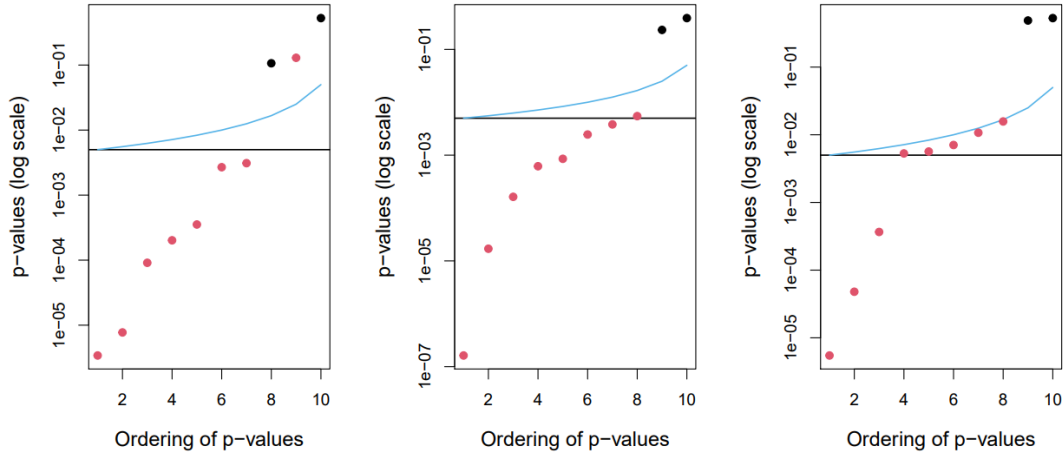


FIGURE 5 – Holm vs Bonferroni

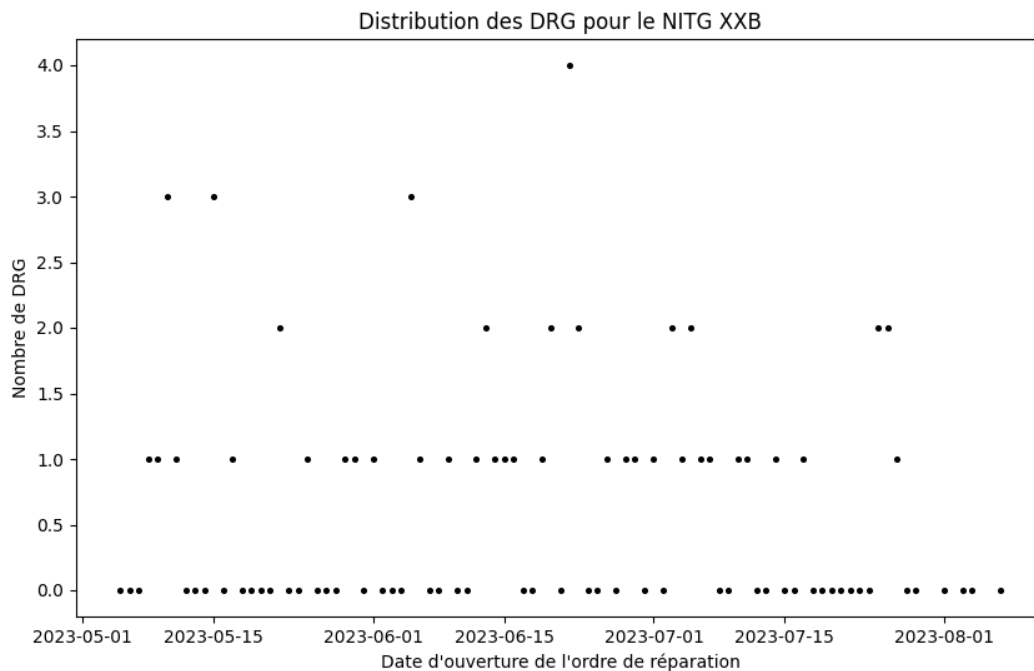
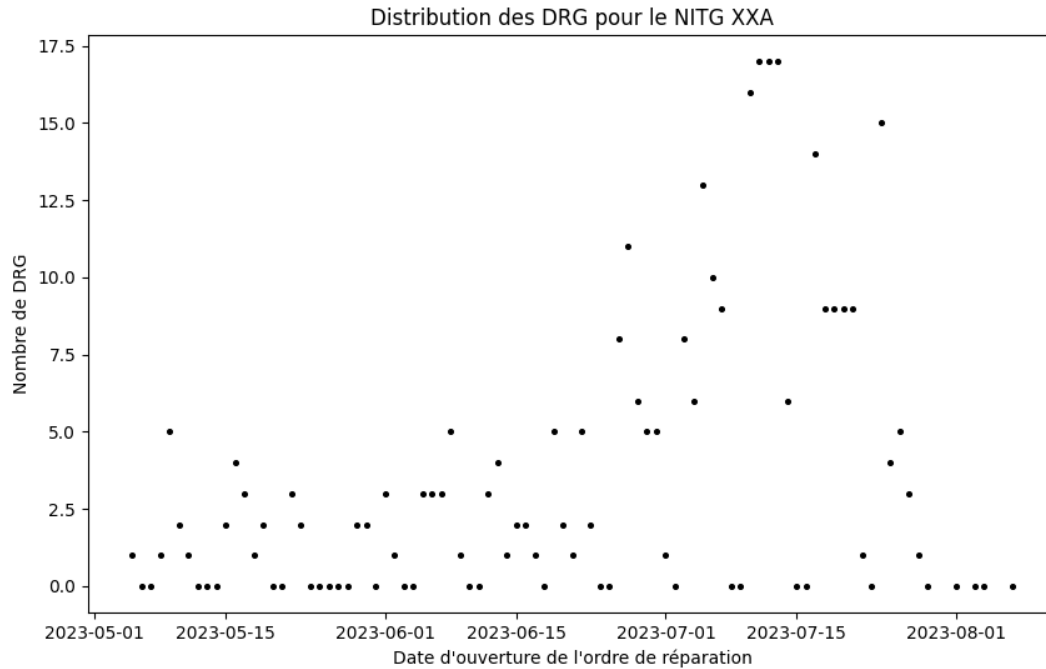
Chacun de ces graphes représente une simulation d'un processus de tests multiples. Chacun représente les p-valeurs ordonnées des tests de $m = 10$ hypothèses nulles. Le nombre de vraies hypothèses nulles est $m_0 = 2$, les p-valeurs correspondantes sont représentées en noir, les autres sont représentées en rouge. En contrôlant le FWER au niveau $\alpha = 0.05$, la procédure de Bonferroni rejette toutes les hypothèses nulles en dessous de la ligne horizontale noire et la procédure de Holm rejette toutes les hypothèses nulles en dessous de la courbe bleue. La région entre ces deux courbes représente les hypothèses nulles rejetées par la procédure de Holm mais non rejetées par la procédure de Bonferroni. Nous pouvons remarquer que sur le graphe central il y a une fausse hypothèse nulle non rejetée par Bonferroni mais rejetée par Holm (à raison) et sur le graphe de gauche il y'en a 5. Cela illustre bien l'aspect moins conservateur de la procédure de Holm comparée à celle de Bonferroni.

D'autres procédures plus puissantes existent mais elles supposent que les p-valeurs soient indépendantes ou positivement corrélées, ce qui n'est pas le cas ici.

En résumé, les deux procédures sont conçues pour contrôler l'erreur de type I globale dans les tests statistiques multiples et ne nécessite aucune hypothèses sur les tests. La procédure de Holm est plus puissante et moins conservatrice que la procédure de Bonferroni.

2.2.5 Illustrations sur données simulées

Nous analysons la distribution des DRG associées au NITG 'XXA' et 'XXB' sur la période allant du 01 mai 2023 au 31 août 2023 en considérant la date d'ouverture de l'ordre de réparation.



Nous utilisons l'algorithme de scan statistic avec les tailles de fenêtre multiples. Nous utilisons la procédure de test de Holm et nous testons les tailles de fenêtre suivantes : [5, 10, 15, 20, 25, 30].

Nous obtenons les résultats suivant :

- Le NITG 'XXA' est suspect.
- La taille de cluster est de 30 jours.
- Le cluster s'étend du 25 Juin 2023 au 25 Juillet 2023.

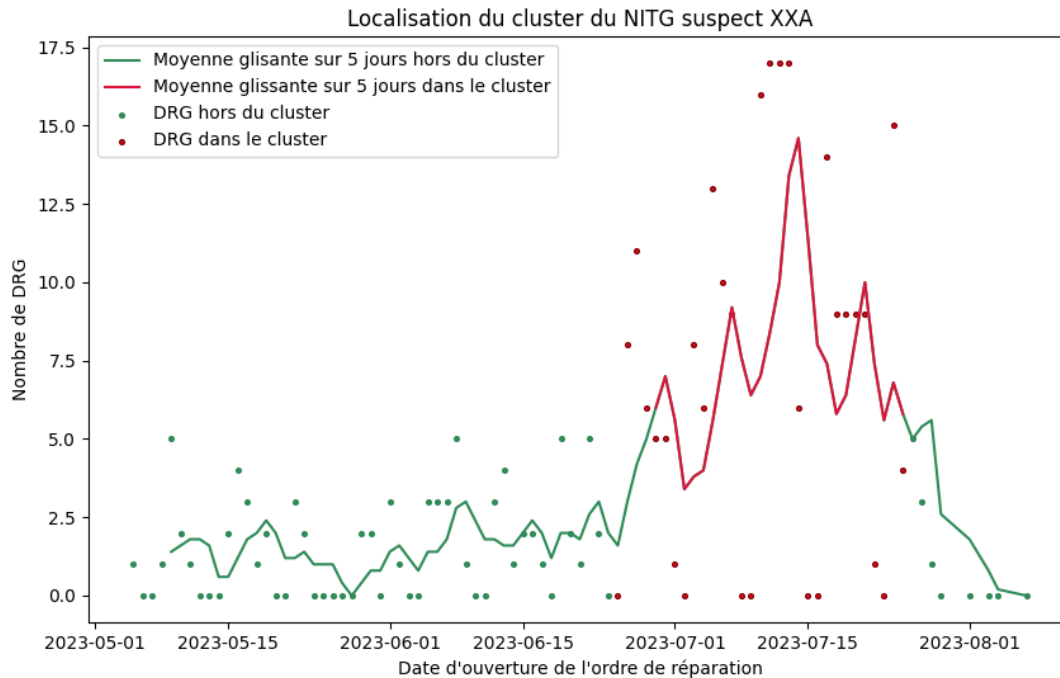
P-valeur	Borne Test Holm	Taille fenêtre	S_w
6.1 e-42	8.3 e-3	30	225
4.0 e-36	1.0 e-2	25	197
3.3 e-33	1.25 e-2	20	171
4.2 e-28	1.7 e-2	15	139
1.7 e-23	2.5 e-2	10	105
7.5 e-23	5.0 e-2	5	73

TABLE 2 – Tableau récapitulatif de la procédure de tests multiples avec correction de Holm du NITG XXA

- Le NITG 'XXB' n'est pas suspect, il n'y a pas de cluster.
- Le plus petite p-valeur de la procédure de tests multiples est déjà non significative. L'hypothèse nulle n'est jamais rejetée quelque soit la taille de fenêtre considérée.

P-valeur	Borne Test Holm	Taille fenêtre	S_w
0.104	0.008	25	25

Nous pouvons visualiser les résultats de l'analyse du NITG XXA sur le graphique ci-dessous. La moyenne glissante sur 5 jours permet de mieux visualiser l'évolution du nombre de DRG au cours du temps.



En résumé, les scan statistics avec des tailles de fenêtre multiples présentent une réelle amélioration par rapport aux scan statistics avec fenêtre fixe. Elles permettent de ne plus être sujet au problème de sensibilité lié au choix de la taille de la fenêtre tout en contrôlant le FWER. Ainsi, des clusters de DRG de tailles variables peuvent être détectées sans augmenter le nombre de fausses détections.

2.3 Scan statistics sur variables continues

2.3.1 Idée Générale

Nous avons analysé grâce aux scan statistics la distribution du nombre de DRG (demandes de remboursements garantis) dans le temps. S'agissant de données discrètes, l'occurrence des événements a été modélisée par un processus de poisson. Cette modélisation nous a permis d'avoir une formule approximative de la probabilité $\mathbb{P}(S_\omega \geq k)$ d'observer un cluster de taille k ou plus et ainsi de construire un test de taille α . Lorsque que nous souhaitons analyser la distribution des coûts, nous sommes confrontés au problème de la modélisation. En effet, lorsque nous observons la distribution des coûts au cours du temps, aucune distribution connue ne peut être attribuée aux données. Nous ne pouvons pas calculer $\mathbb{P}(S_\omega \geq k)$ et construire une procédure de test de niveau α .

La solution est de construire un test non paramétrique et de mesurer la significativité avec un test randomisé modifié de Monte Carlo, comme cela est fait dans le papier de Kuldorff, Huang et Konty. [7].

Dans ce papier, une scan statistic pour des données continues a été développée en se basant sur la loi normale.

Sous l'hypothèse nulle, toutes les observations proviennent de la même distribution. Sous l'hypothèse alternative, il existe un cluster où les observations ont des valeurs plus grandes ou plus petites qu'à l'extérieur du cluster. Pour notre application nous ne considérerons que les clusters avec des valeurs plus grandes.

Cette méthode est particulièrement intéressante car l'inférence statistique est toujours valide même si la vraie distribution des données ne suit pas une loi normale. Cela s'explique par le fait que la significativité statistique du cluster est évaluée par une p-valeur estimée par des permutations des données suivies d'une procédure de Monte Carlo.

2.3.2 Formalisation

Les données de coûts sont des réalisations d'une variable aléatoire continue. Chaque observation est indépendante des autres. Soit N le nombre total d'observations et (x_1, x_2, \dots, x_N) les coûts observés.

Nous considérons une fenêtre temporelle glissante z caractérisée par sa localisation et sa taille. Pour chaque fenêtre z , nous allons calculer le log rapport de vraisemblance entre le modèle avec cluster (M_z , de vraisemblance L_z) et le modèle sans cluster (M_0 , de vraisemblance L_0), nous le notons $LLR(z)$.

Nous notons :

- X la somme totale des coûts.
- x_z la somme des coûts dans la fenêtre z .
- n_z le nombre d'observations dans la fenêtre z .

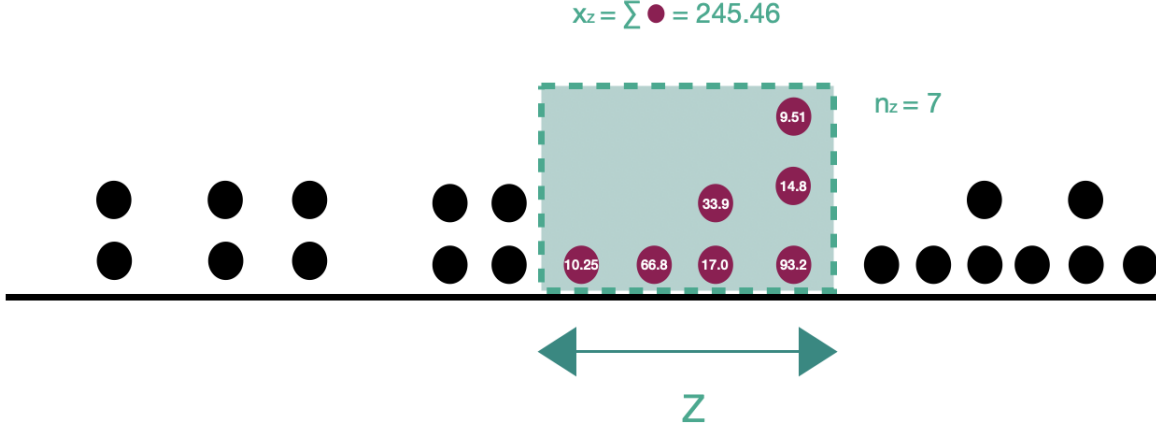


FIGURE 6 – Représentation d’une distribution de coûts. Le rectangle vert représente la fenêtre z .

— **Sous l’hypothèse nulle :**

- L’estimateur du maximum de vraisemblance pour la moyenne est X/N et celui de la variance est $\sigma = \frac{1}{N} \sum_i (x_i - \mu)^2$.
- La log vraisemblance s’écrit

$$\ln L_0 = -N \ln(\sqrt{2\pi}) - N \ln(\sigma^2) - \sum_i \frac{(x_i - \mu)^2}{2\sigma^2}$$

— **Sous l’hypothèse alternative :**

- Les estimateurs de la moyenne sont calculés séparément. Nous obtenons à l’intérieur de la fenêtre z , $\mu_z = x_z/n_z$, et à l’extérieur de la fenêtre, $\lambda_z = (X - x_z)/(N - n_z)$.
- L’estimateur du maximum de vraisemblance de la variance commune est donné par

$$\sigma_z^2 = \frac{1}{N} \left(\sum_{i \in z} x_i^2 - 2x_z \mu_z + n_z \mu_z^2 + \sum_{i \notin z} x_i^2 - 2(X - x_z) \lambda_z + (N - n_z) \lambda_z^2 \right)$$

- La log vraisemblance s’écrit

$$\ln L_z = -N \ln(\sqrt{2\pi}) - N \ln(\sqrt{\sigma_z^2}) - N/2$$

Le test statistique est défini comme le maximum du log rapport de vraisemblance sur les configurations z de la fenêtre glissante.

$$T = \max_z LLR(z) \tag{1}$$

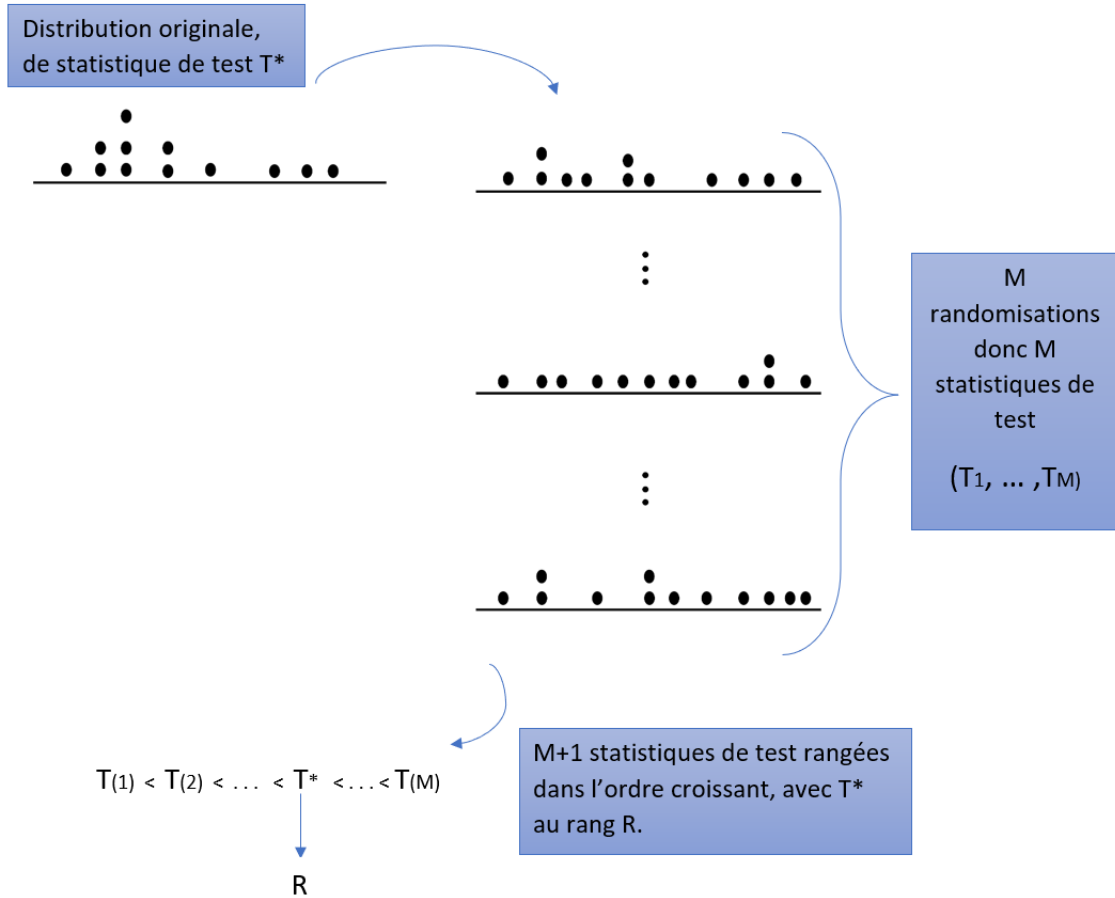
$$= \max_z \left(\frac{\ln L_z}{\ln L_0} \right) \tag{2}$$

$$= \max_z \left(N \ln(\sigma) + \sum_i \frac{(x_i - \mu)^2}{2\sigma^2} - \frac{N}{2} - N \ln(\sqrt{\sigma_z^2}) \right) \tag{3}$$

Le seul terme qui dépend de z est σ_z , la variance sous l’hypothèse alternative. Le cluster le plus probable est donc celui qui minise σ_z .

Pour mesurer la significativité de la statistique de test T , la p-valeur du test va être estimée par la méthode de Monte Carlo. Les données sont aléatoirement changées de localisation temporelle. Nous notons R le classement relatif de la statistique de test des données originales parmi les valeurs des statistiques de test obtenues sur les échantillons des randomisations de Monte Carlo. Soit M le nombre de randomisations de Monte Carlo réalisées. La p-valeur du cluster le plus vraisemblable est donnée par :

$$P = \frac{R}{M + 1}$$



Remarque 1 Cette méthode de calcul de la p-valeur peut s'avérer peu robuste. Lorsque les statistiques de test atteignent le même minimum, il est possible de se retrouver avec plusieurs valeurs identiques. Si la statistique de test de la distribution originale T^* en fait partie, le choix du rang R (première occurrence ou dernière occurrence de la valeur) change la p-valeur et influence le résultat du test. Une solution à ce problème est de lisser la fonction de répartition empirique des p-valeurs à l'aide de technique d'interpolation. Cette amélioration n'a pas encore été apportée à l'algorithme .

2.3.3 Illustrations sur données simulées

Nous avons simulé deux distributions de coûts. Une distribution globalement uniforme que nous allons attribuer au NITG XXC et une distribution de coûts présentant un cluster que nous allons attribuer au NITG XXD.

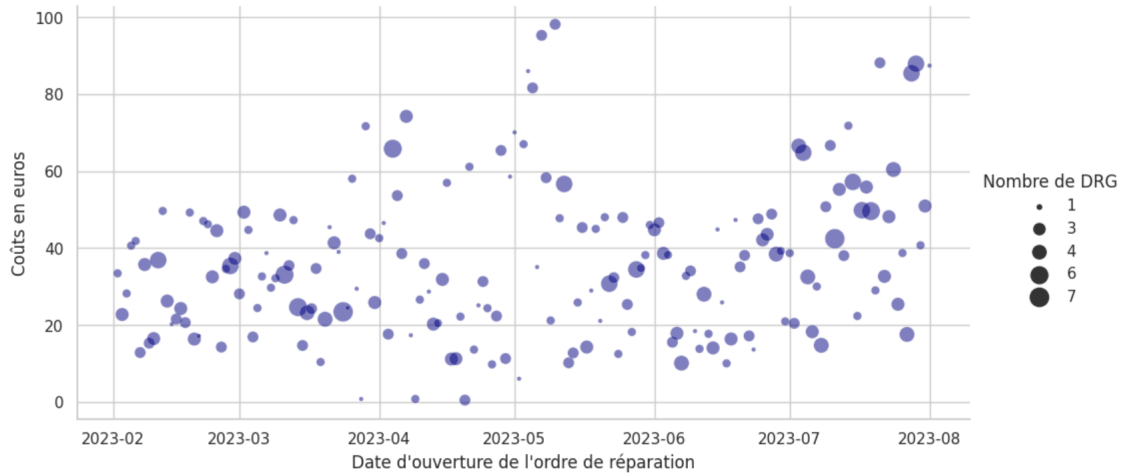


FIGURE 7 – Distribution des coûts des DRG associées au NITG XXC

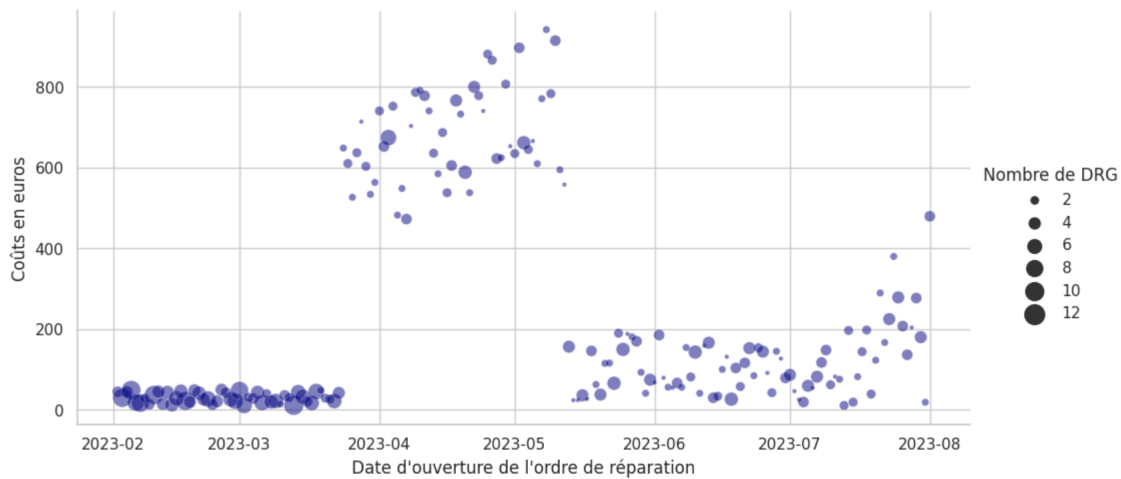


FIGURE 8 – Distribution des coûts des DRG associées au NITG XXD

Nous choisissons de tester 7 tailles de fenêtre, 5,10,15,20,25,30 et 35 jours. Nous faisons 999 randomisations de Monte Carlo. Nous obtenons les résultats suivants :

Pour le NITG XXC :

- P-valeur = 0.612
- Suspect = Non (Hypothèse nulle non rejetée)
- Taille du cluster le plus vraisemblable = 5 jours
- Nombre de DRG dans le cluster = 6
- Coût total dans le cluster = 287.60

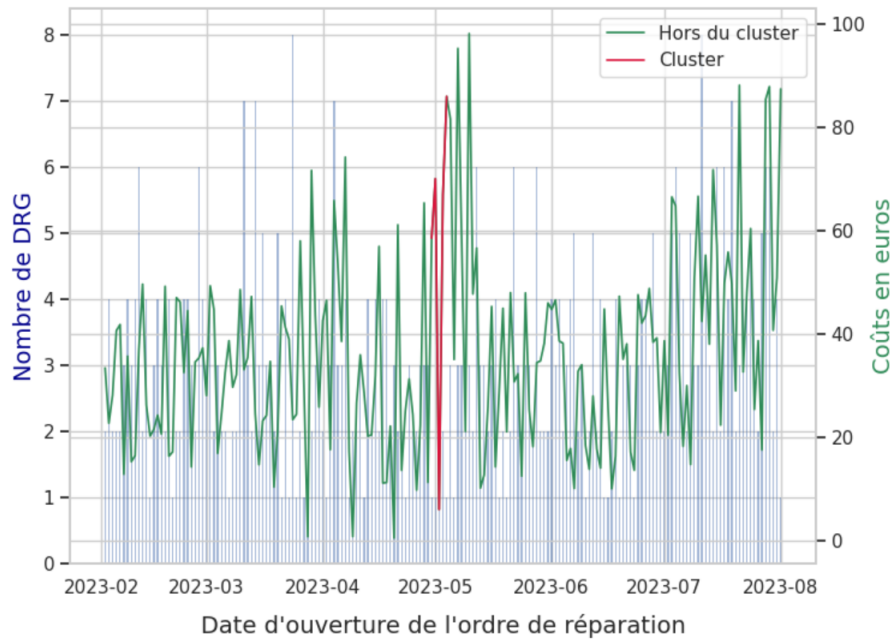


FIGURE 9 – Ce graphique montre le nombre et le coût des DRG associées au NITG XXC au cours du temps, **le cluster le plus vraisemblable** est en rouge.

La figure 9 n'est pas la plus pertinente pour comprendre la localisation du cluster le plus vraisemblable. En effet, sur ce graphique, c'est le coût total quotidien qui est représenté. Dans notre approche, il s'agit plutôt du coût moyen par fenêtre glissante qui est étudié. Nous ne cherchons pas à détecter des clusters sur les coûts totaux. Lorsque nous détectons les clusters de DRG (ce que nous faisons avec les scan statistics sur les données discrètes) nous avons déjà des informations sur les coûts totaux. En effet, plus il y a de DRG, plus le coût total est important.

Le graphique 10 nous montre l'évolution du coût moyen dans une fenêtre glissante de la taille du cluster le plus vraisemblable, soit 5 jours. Cette représentation permet de mieux comprendre la localisation du cluster le plus vraisemblable.

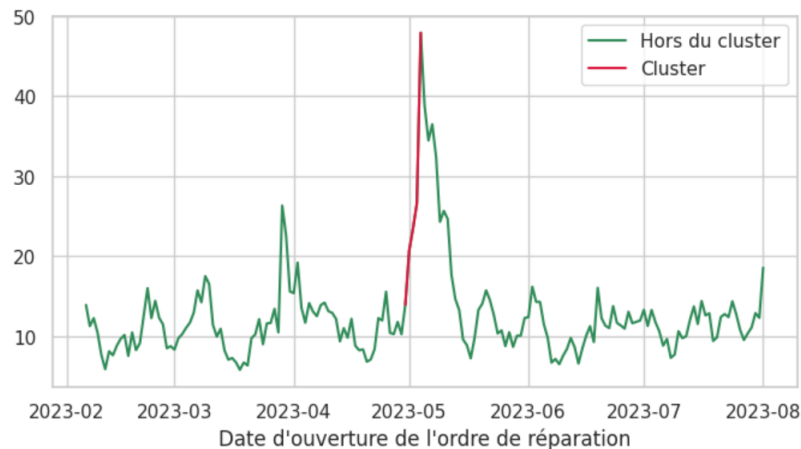


FIGURE 10 – Moyenne glissante des coûts sur une fenêtre de la taille du cluster soit 5 jours pour le NITG XXC. **Le cluster le plus vraisemblable** est en rouge

Pour le NITG XXD :

- P-valeur = 0.016
- Suspect = Oui (Hypothèse nulle rejetée)
- Taille du cluster le plus vraisemblable = 35 jours
- Nombre de DRG dans le cluster = 100
- Coût total dans le cluster = 24 917.72

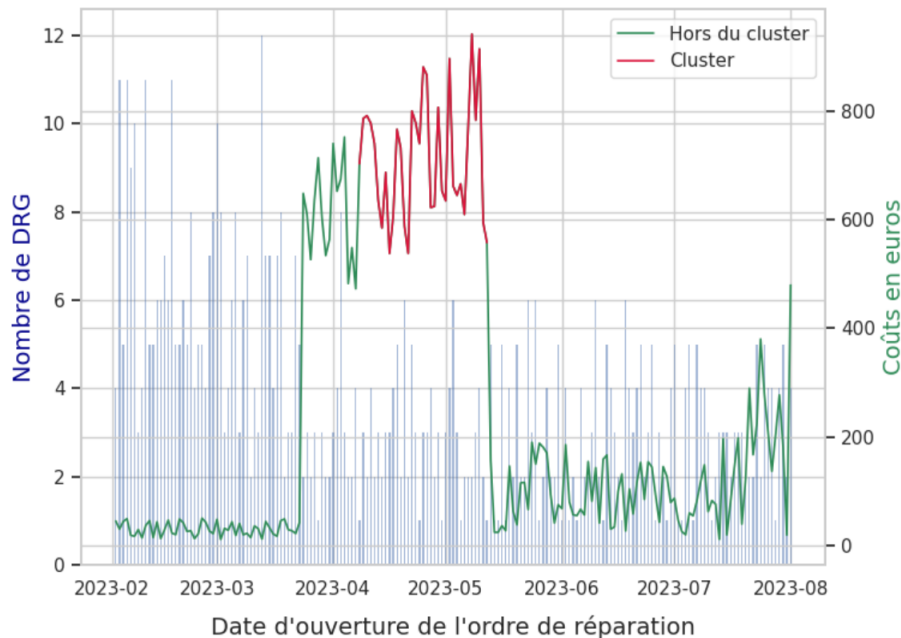


FIGURE 11 – Ce graphique montre le nombre et le coût des DRG associées au NITG XXD au cours du temps, **le cluster le plus vraisemblable** est en rouge.

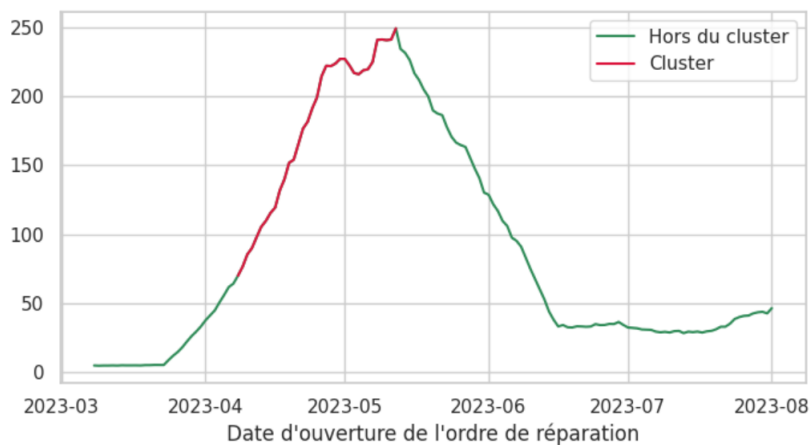


FIGURE 12 – Moyenne glissante des coûts sur une fenêtre de la taille du cluster soit 35 jours pour le NITG XXD. **Le cluster le plus vraisemblable** est en rouge.

3 Maintenance prédictive des Presses d'emboutissage des usines

Les presses d'emboutissage sont des machines utilisées pour façonner des pièces de carrosserie en métal, comme des panneaux de carrosserie ou des éléments de châssis, par déformation à froid.

La maintenance prédictive des presses d'emboutissage des usines Renault est une approche proactive qui permet la gestion de la maintenance de ses équipements critiques. Cette technique repose sur la collecte et l'analyse de données en temps réel ou à intervalles réguliers, généralement à l'aide de capteurs et de systèmes de surveillance.

Les capteurs sont installés sur les presses d'emboutissage pour collecter une variété de données, telles que les vibrations, la température, la pression, le courant électrique, et d'autres paramètres liés au fonctionnement de la machine.

La maintenance prédictive vise à anticiper les défaillances potentielles de ces presses en se basant sur des analyses avancées.

3.1 Présentation du sujet

Actuellement, la surveillance des machines des usines se fait dans un espace centralisé où les experts surveillent la bonne collecte des données par les capteurs et analysent les signaux.

Ce travail peut vite devenir fastidieux étant donné le nombre important d'usines Renault dans le monde et le nombre de machines à surveiller.

Notre objectif est de faciliter le travail de surveillance, tout d'abord en créant un indicateur de l'état de santé des machines, et ensuite en facilitant l'identification du problème à l'origine de la défaillance.

Nous cherchons à analyser les données collectées à l'aide d'algorithmes et de techniques d'intelligence artificielle pour détecter les signes précurseurs de problèmes potentiels, tels que des variations dans les vibrations ou des températures anormalement élevées. En se basant sur ces analyses, nous devons créer un outil capable de prédire quand une presse d'emboutissage pourrait présenter un risque de défaillance ou nécessiter une maintenance. Sur la base des prédictions, le personnel de maintenance peut planifier des interventions de maintenance préventive ou corrective au moment opportun, avant qu'une panne majeure ne survienne. Cela permet de minimiser les temps d'arrêt imprévus de la production et d'optimiser l'efficacité de l'usine.

3.2 Description des données

Le problème auquel nous avons été confronté et celui de la qualité des données. Nous disposons d'un dataset étalon et d'un fichier d'alertes dans lequel était répertorié les incidents survenus sur les machines. Nous avons observé les problèmes suivants :

- Certaines données d'acquisition étaient manquantes.
- Les périodes d'acquisition n'étaient pas synchronisées pour certains capteurs.
- La période d'acquisition était trop courte, certains signaux n'avaient des acquisitions que sur une période qui correspondait à une alerte. Nous n'avons pas d'acquisition correspondant à un état de la machine en bonne santé.
- Le fichier des données était mal organisé et la nomenclature des fichiers pas toujours respectée. Ce

qui rendait l'identification des signaux compliquée.

- Les données provenaient de 2 usines seulement.

Après une visite à la salle de contrôle, nous avons pu choisir des données plus exploitables pour pouvoir entraîner un algorithme de machine learning.

Notre dataset est composé de signaux provenant de 6 usines (Tanger, Pitesti, Valladolid, Curitiba, Flins, Palencia) sur une durée de 2 à 5 ans où s'enchaînent des périodes sans problème particuliers et des périodes de défaillances identifiées par les experts de la salle de contrôle.

3.3 Analyse des données et approche envisagée

Avant de constituer notre propre dataset, nous ne disposions que du dataset étalon qui correspond aux acquisitions des presses de 2 usines. Ce sont sur ces données que nous avons pu faire nos premières explorations. Nous avons pu remarquer que :

- Les périodes d'alerte sont précédées d'une tendance ascendante des signaux pour certains indicateurs.
- Il existe un seuil d'alerte et un seuil de danger pour chaque indicateur. Ces seuils sont choisis et contrôlés régulièrement par les fournisseurs des machines.
- Dépasser le seuil d'alerte n'est pas systématiquement le signe avant-coureur d'une défaillance. L'approche simpliste qui consiste à envoyer une alerte dès qu'un seuil est dépassé n'est pas idéale car elle mènerait à beaucoup de fausses alertes.
- Les indicateurs sont corrélés les uns avec les autres. Il est important de croiser les informations données par plusieurs indicateurs pour identifier la défaillance ou pour s'assurer qu'il s'agit bien d'une défaillance machine et non d'une défaillance du capteur ou du système d'enregistrement des acquisitions.

En échangeant avec les experts de la salle de contrôle, nous avons compris qu'il était nécessaire de :

- Créer un modèle d'une assez bonne interprétabilité pour que les décisions prises soient éclairées.
- Contrôler les fausses alertes autant que possible, pour que l'outil ait un réel intérêt.
- S'inspirer des méthodes actuelles d'analyse pour créer des indicateurs pertinents.

L'approche qui a été envisagée est la suivante :

- Découper les signaux en intervalles de temps réguliers, annoter chaque intervalle en indiquant s'il correspond à une période d'alerte ou non.
- Construire des indicateurs pertinents. Pour chaque intervalle, calculer ces indicateurs puis procéder à une sélection de variable.
- Entraîner un algorithme de classification avec les variables sélectionnées et construire un indicateur de l'état de santé de la machine. Dans le cadre d'un apprentissage supervisé, nous pouvons construire un modèle de régression logistique et utiliser la sortie de la fonction logistique comme score de l'état de santé de la machine.

3.4 Construction d'indicateurs pertinents

Afin de construire un modèle de classification, il est nécessaire de construire des variables pertinentes permettant au modèle de prendre en compte les caractéristiques les plus importantes des données, ce qui peut améliorer sa capacité à discriminer entre les différentes classes.

Pour cela nous nous sommes inspirés de la méthode utilisée par les experts de la salle de contrôle. Sur

le tableau de bord, ils observent les trajectoires des signaux de certains indicateurs clés. Un code couleur permet de savoir si un seuil a été dépassé. Si l'indicateur est vert, aucun seuil n'a été dépassé, s'il est jaune le seuil d'alerte a été dépassé et s'il est rouge le seuil de danger a été dépassé. Si un seuil a été dépassé, les experts regardent si la tendance de la courbe est à la hausse et la surveillance sur plusieurs jours pour savoir si le problème est persistant.

C'est cette démarche que nous avons essayé de reproduire pour créer une nouvelle variable. Notre objectif est de détecter les ruptures dans les signaux et calculer le coefficient des pentes pour chaque intervalle. Ainsi nous aurons une nouvelle variable pertinente pour notre modèle.

3.4.1 Point de rupture et modèle de rupture dans la tendance

Nous considérons l'espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$. Comme nous manipulons des données qui appartiennent à \mathbb{R} , posons $\Omega = \mathbb{R}$ et $\mathcal{F} = \mathcal{B}(\mathbb{R})$, la tribu borélienne.

Dans notre cas, pour chaque indicateur nous assimilons les acquisitions à un processus stochastique $\{Y_t : t \in \mathbb{N}\}$. Nous introduisons les notations suivantes que nous utiliserons par la suite.

Soit $y = (y_1, \dots, y_n)$ une observation du processus $Y = (Y_1, \dots, Y_n)$, pour tout $\tau \in \llbracket 1, n \rrbracket$ on note $y_{1:\tau}$ la sous série (y_1, \dots, y_τ) . De même pour tout $\tau_1, \tau_2 \in \llbracket 1, T \rrbracket$ tel que $\tau_1 < \tau_2$, $y_{\tau_1:\tau_2} = (y_{\tau_1}, \dots, y_{\tau_2})$.

Tout d'abord définissons ce qu'est un point de rupture.

Définition 1 *Un point de rupture est un moment où les propriétés statistiques de la série que l'on observe changent. Soit une séquence ordonnée de données $x_{1:n}$, on dit qu'il y a un point de rupture dans la série s'il existe $\tau \in \llbracket 1, \dots, n-1 \rrbracket$ tel que les propriétés statistiques de $x_{1:\tau}$ et $x_{\tau+1:n}$ soient différentes.*

Nous notons K le nombre de points de rupture. Nous noterons τ_1, \dots, τ_K les K points de rupture.

Définissons $\llbracket 0, \tau_1 \rrbracket$ le segment 0, $\llbracket \tau_k + 1, \tau_{k+1} \rrbracket, k = 1, \dots, K-1$ le segment k et $\llbracket \tau_K + 1, n \rrbracket$ le segment K . Posons $(z_t)_t$ un bruit blanc iid de variance supposée finie tel que $z_t \sim \mathcal{N}(0, \sigma^2)$.

Nous nous intéressons au modèle de rupture dans la tendance. Sur le k -ième segment du processus, la série sera approchée par une droite qui lui est propre de la forme $\mu_k + \beta_k t$, où μ_k est appelé **intercept** (ou ordonnée à l'origine) et β_k est la **tendance** de la droite (ou le coefficient directeur). Le modèle de rupture dans la tendance se définit alors comme suit :

$$y_t = \begin{cases} \mu_0 + \beta_0 t + z_t, & 1 \leq t \leq \tau_1 \\ \mu_1 + \beta_1 t + z_t, & \tau_1 + 1 \leq t \leq \tau_2 \\ \vdots \\ \mu_K + \beta_K t + z_t, & \tau_K + 1 \leq t \leq n \end{cases}$$

Une fois le modèle choisi, nous devons définir une fonction de coût. La fonction de coût nous permet de quantifier l'erreur entre les données empiriques et les estimations fournies par notre modèle. Ainsi, nous avons pour objectif d'avoir le modèle avec le coût le plus faible possible. La fonction de coût doit donc être minimisée. Nous utilisons la fonction de coût quadratique.

3.4.2 L'algorithme PELT

L'algorithme PELT (Pruned exact linear time) est l'algorithme de résolution que nous allons utiliser. Il présente plusieurs avantages, notamment sa complexité linéaire, qui fait de lui l'un des algorithmes de résolution les plus rapides, et son adéquation au cas où le nombre de points de rupture est inconnu.

L'algorithme PELT est un algorithme de détection de points de rupture adapté aux problèmes où le nombre de point de rupture est inconnu. L'objectif est de minimiser le coût et la pénalité liée au fait d'avoir k points de rupture :

$$\underbrace{\sum_{i=1}^{k+1} \left[\mathcal{C}(y_{\tau_{i-1}+1: \tau_i}) \right]}_{\text{le coût}} + \underbrace{\text{pen}(k)}_{\text{la pénalité}}$$

L'algorithme PELT a été conçu pour une fonction de pénalité linéaire. C'est-à-dire les fonctions de la forme :

$$\mathcal{P} : \begin{cases} \mathbb{N} & \longrightarrow \mathbb{R} \\ k & \longmapsto \beta(k+1) \end{cases}$$

où $\beta \in \mathbb{R}^+$

On obtient donc :

$$\sum_{i=1}^{k+1} \left[\mathcal{C}(y_{\tau_{i-1}+1: \tau_i}) \right] + \beta(k+1)$$

En introduisant le terme de pénalité dans la somme, nous obtenons :

$$\sum_{i=1}^{k+1} \left[\mathcal{C}(y_{\tau_{i-1}+1: \tau_i}) + \beta \right]$$

Pour comprendre le principe de l'algorithme PELT, nous allons introduire :

- K_{max} le nombre maximal de points de rupture.
- $\mathcal{T}_t = \{\tau_{0:K+1} : \tau_0 < \tau_1 < \dots < \tau_K < \tau_{K+1} = t, K = 1, \dots, K_{max}\}$, l'ensemble des configurations possibles de K points de rupture pour $K = 1, \dots, K_{max}$ de la série $y_{1:t}$.
- $F(s) = \min_{\tau \in \mathcal{T}_s} \sum_{k=1}^{K_{opt}+1} [\mathcal{C}(y_{\tau_{k-1}+1: \tau_k}) + \beta]$. On note K_s^{opt} le nombre de points de rupture qui permet d'atteindre ce minimum.

Soit $t \in [0, s]$:

$$F(s) = \min_{\tau \in \mathcal{T}_s} \sum_{k=1}^{K_s^{opt}+1} [\mathcal{C}(y_{\tau_{k-1}+1: \tau_k}) + \beta] \quad (4)$$

$$= \min_t \left\{ \min_{\tau \in \mathcal{T}_t} \sum_{k=1}^{K_t^{opt}} [\mathcal{C}(y_{\tau_{k-1}+1: \tau_k}) + \beta] + \mathcal{C}(y_{(t+1):s}) + \beta \right\} \quad (5)$$

$$= \min_t \{ F(t) + \mathcal{C}(y_{(t+1):s}) + \beta \} \quad (6)$$

Donc s'il existe un t tel que :

$$F(t) + \mathcal{C}(y_{(t+1):s}) + \beta > F(s)$$

Alors t ne peut pas être le dernier point de rupture optimal de la série $y_{1:s}$. Grâce à ce constat, on peut donc éliminer les points de rupture non optimaux et ne garder que les points de rupture qui peuvent potentiellement être optimaux, nous allons les appeler points de rupture admissibles. C'est le principe d'élagage de l'algorithme PELT. C'est ce principe qui lui permet d'avoir une complexité linéaire.

3.4.3 Application

Pour l'application, nous utilisons le package "ruptures" [8], il s'agit d'une bibliothèque dédiée à la détection de changements structurels dans des séries temporelles ou des données univariées. Il permet d'identifier les points où une série de données présente un changement significatif dans sa distribution ou sa structure.

Le package rupture offre plusieurs algorithmes de détection de ruptures prédéfinis, tels que la rupture de variance, la rupture de pente, etc. Il permet également aux utilisateurs de définir leurs propres critères de rupture personnalisés.

Etant donné que nous souhaitons détecter les changements de tendance dans les signaux, c'est le modèle de rupture dans la tendance du package que nous devrions utiliser. Cependant, lors de l'application nous nous sommes rendus compte que les détections avec le modèle de rupture dans la tendance du package n'étaient pas cohérentes, ce problème est probablement dû à une mauvaise utilisation du package "ruptures". Nous avons finalement choisi d'utiliser un modèle basé sur les méthodes à noyaux ; une méthode que je n'ai pas pu étudier de façon approfondie pour en comprendre le fonctionnement faute de temps. Ce modèle permet quand même de détecter les changements de pente de la série étudiée. Nous choisissons l'algorithme PELT comme algorithme de résolution pour ses avantages décrits dans la section précédente.

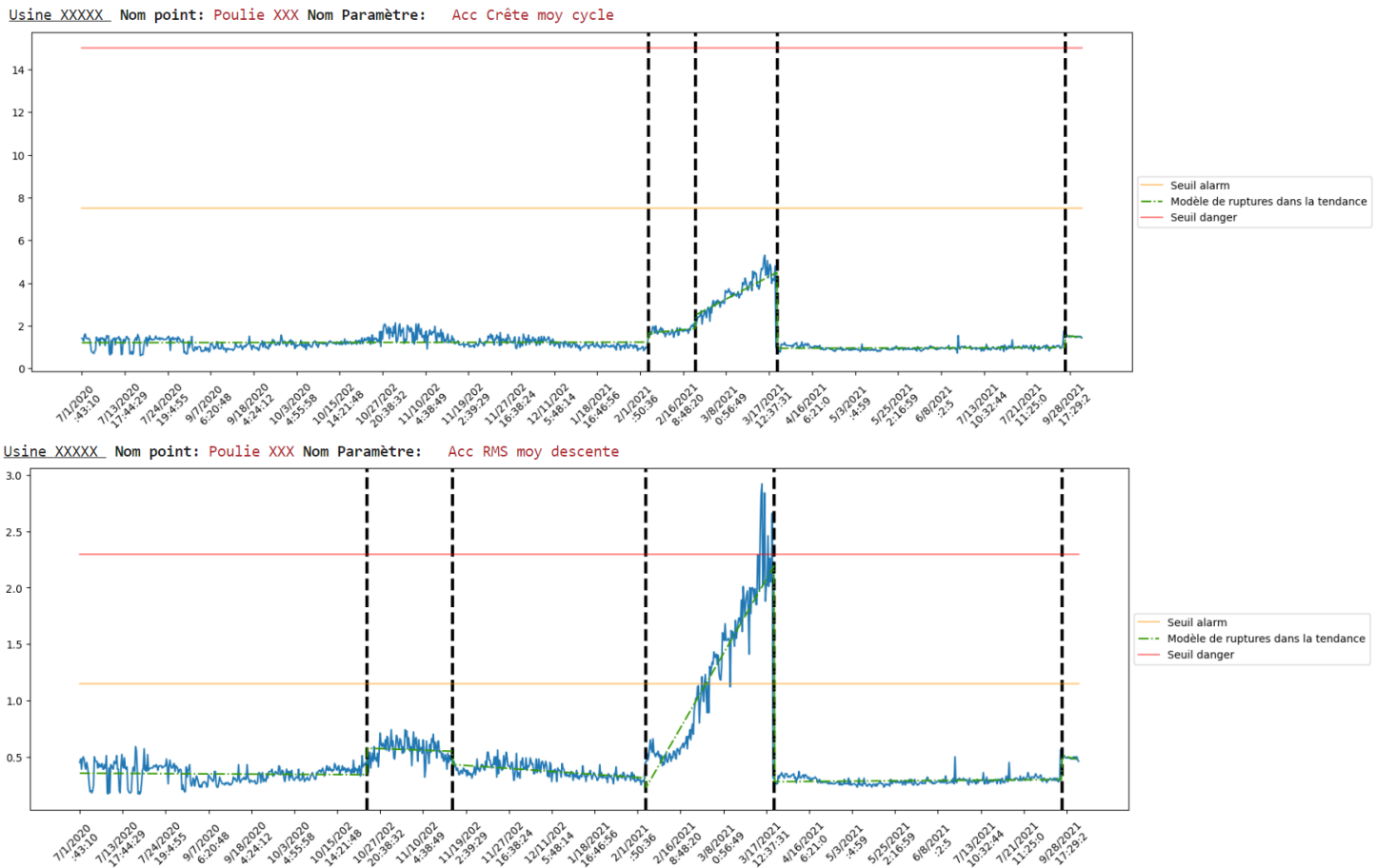


FIGURE 13 – Détection de point de rupture sur acquisition de deux paramètres de la poulie de l'usine XXXXX.

Malheureusement, en raison de contraintes de temps, je n'ai pas pu procéder à la sélection de variable et entraîner le modèle de régression logistique sur les nouvelles variables créées.

Conclusion

Le stage en data science au sein du Groupe Renault a été une expérience extrêmement enrichissante et instructive. L'objectif principal de ce stage était d'appliquer les compétences en data science acquises au cours de ma formation et de les mettre en pratique dans un environnement professionnel. Je suis ravie de partager les principales conclusions et réalisations de cette expérience.

Au cours de ce stage, j'ai eu l'opportunité de travailler sur divers projets, liés à la détection de dérives sur les demandes de remboursements garantis, au sein de la direction de la qualité . J'ai pu participer à l'automatisation de l'outil de scan statistic existant, permettant ainsi la mise à jour hebdomadaire des alertes et facilitant ainsi le travail des incidentologues. J'ai développé une extension de cet outil permettant de limiter la sensibilité à un paramètre du modèle qui est la taille de la fenêtre d'analyse. Pour finir, j'ai implémenté un outil de scan statistic adapté aux données continues pour analyser la distribution des coûts, permettant ainsi d'analyser les demandes de remboursements garantis sous un autre angle. Du côté de la manufacturing, sur le sujet de la maintenance prédictive des presses d'emboutissage, j'ai formalisé le problème, proposé une approche et commencé l'implémentation d'une solution . Ainsi, j'ai pu développer des modèles statistiques pour résoudre des problèmes concrets de l'entreprise.

Ces projets m'ont permis de consolider mes compétences en programmation, en manipulation de données et en modélisation statistique. Même si certains n'ont pas pu être abouti, je pense que ma contribution facilitera la poursuite de ces projets.

J'ai également eu l'occasion de travailler en étroite collaboration avec plusieurs métiers et domaines d'expertise, notamment les experts de maintenance à la manufacturing, les incidentologues, statisticiens et data scientists au sein de la direction de la qualité et les développeurs dans le cadre de l'automatisation de l'outil de scan statistics. Le partage de connaissances et d'idées au sein de l'équipe a été une expérience stimulante et m'a permis d'apprendre de nouvelles techniques et approches.

Cependant, il est important de noter que ce stage n'a pas été sans défis. Les données brutes que nous avons traitées étaient souvent incomplètes ou mal organisées, ce qui a nécessité des efforts importants pour les nettoyer et les préparer pour l'analyse. J'ai également dû présenter mon approche et mes résultats à un public qui n'a pas forcément des connaissances avancées en modélisation statistique. Ces défis m'ont permis d'améliorer mes compétences.

En conclusion, ce stage en data science au sein du groupe Renault a été une étape cruciale dans mon développement professionnel. J'ai pu appliquer mes compétences théoriques à des problèmes réels, acquérir de nouvelles compétences et collaborer avec une équipe inspirante. Je suis reconnaissante envers toute l'entreprise pour cette opportunité .

Références

- [1] J. Naus, *Clustering of Random Points in Line and Plane*. Harvard University, 1963. **url:** <https://books.google.fr/books?id=uW00GwAACAAJ>.
- [2] J. Glaz, J. Naus **and** S. Wallenstein, *Scan Statistics* (Springer series in statistics). Springer, 2001.
- [3] J. Glaz, X. Wang **and** B. Zhao, “A multiple window scan statistic for time series models,” *Statistics and Probability Letters*, **number** 9, **pages** 196–203, 2014.
- [4] J. Glaz, X. Wang **and** B. Zhao, “A new adaptive procedure for multiple window scan statistics,” *Computational Statistics and Data Analysis*, **number** 82, **pages** 164–172, 2015.
- [5] M. Kulldorff, F. Mostashari **and** L. Duczmal, “Multivariate scan statistics for disease surveillance,” *Stat Med.*, **number** 1824-33, **april** 2007.
- [6] G. James, D. Witten, T. Hastie, R. Tibshirani **and** J. Taylor, *An Introduction to Statistical Learning with applications in Python*. Springer Cham, 2023.
- [7] M. Kulldorff, L. Huang **and** K. Konty, “A scan statistic for continuous data based on the normal probability model,” *International Journal of Health Geographics*, **number** 8 :58, October 2009.
- [8] C. Truong, L. Oudre **and** N. Vayatis, “Selective review of offline change point detection methods,” *Signal Processing*, **journal** 167, **page** 107 299, 2020, ISSN: 0165-1684. DOI: <https://doi.org/10.1016/j.sigpro.2019.107299>. **url:** <https://www.sciencedirect.com/science/article/pii/S0165168419303494>.
- [9] J. Naus, “Approximations for distributions of scan statistics,” *Journal of the American Statistical Association*, **number** 77, **pages** 177–183, 1982.