



# Projet modèles linéaires: Étude des performances des jeux-vidéos présents sur la plateforme Steam

Auteurs:

Daniel Phan | Nathan Ousalem | Antonin Mouillet | Perle M.S. Ndayizeye | Ibrahim Godje

Sous l'encadrement d'Emmanuelle Gautherat et d'Amor Keziou

<b>Introduction.....</b>	<b>2</b>
1. Fonctionnement et rôle de Steam.....	2
2. Contexte Économique et Statistique.....	2
3. Enjeux statistiques de la plateforme.....	3
<b>Présentation des données.....</b>	<b>3</b>
<b>Présentation des variables.....</b>	<b>4</b>
1. Variable à expliquer.....	4
2. Variables explicatives.....	4
3. Population mère et échantillon.....	5
<b>Exploration initiale des données.....</b>	<b>5</b>
Distribution des variables.....	6
Transformation des données.....	6
<b>Analyse des corrélations:.....</b>	<b>6</b>
Création de la variable rating.....	7
<b>Elaboration d'un modèle.....</b>	<b>8</b>
Modèle de base.....	8
Transformation logarithmique.....	9
Un nettoyage accru des données.....	10
L'existence de sous-populations ?.....	10
Solutions possibles: la segmentation par éditeurs.....	11
Optimisation des modèles.....	11
<b>Diverses approches.....</b>	<b>12</b>
Modèles polynomiaux.....	12
Classification.....	13
<b>Conclusion.....</b>	<b>14</b>
<b>Liens:.....</b>	<b>14</b>
<b>Annexe:.....</b>	<b>16</b>

## Introduction

L'industrie du jeu vidéo connaît une expansion rapide, avec des milliers de nouveaux titres publiés chaque année. En 2024, près de 19 000 jeux ont été ajoutés sur Steam, la principale plateforme de distribution numérique de jeux PC, avec plus de **90 millions d'utilisateurs actifs** mensuellement. Cette abondance de choix pose un défi majeur : comprendre quels facteurs influencent l'engagement des joueurs et la durée moyenne du temps consacré à un jeu.

Dans cette étude, nous nous intéressons à l'analyse statistique des jeux présents sur Steam afin d'identifier les variables déterminantes du temps de jeu moyen d'un titre.

### 1. Fonctionnement et rôle de Steam

Steam occupe aujourd'hui une place centrale dans l'industrie du jeu vidéo, notamment en raison de la forte dématérialisation du marché : on estime qu'en 2023, **65% des ventes** de jeux se faisaient sous format numérique. En tant que plateforme de distribution digitale, Steam permet aux développeurs de publier facilement leurs jeux, et aux joueurs de les acheter, télécharger et utiliser en quelques clics. Elle offre également de nombreuses fonctionnalités communautaires, comme les avis utilisateurs, un système d'amis et de messagerie, des classements et des recommandations personnalisées.

### 2. Contexte économique et statistique

Croissance du marché : Le marché des jeux vidéo est en forte expansion, avec une augmentation constante du nombre de joueurs et de jeux disponibles. Ainsi l'on estime que le nombre de joueurs dans le monde atteint à ce jour **3.4 milliards de personnes**<sup>1</sup>, soit un peu moins de la moitié de la population mondiale.

Explosion des données : Steam génère un volume massif de données, incluant le nombre de joueurs actifs, les avis laissés sur les jeux, les tendances de prix et les performances commerciales.

Recommandations et algorithmes : Steam utilise des modèles statistiques pour recommander des jeux aux utilisateurs, en fonction de leurs préférences et de leurs comportements d'achat.

---

<sup>1</sup> le nombre de personnes qui déclare avoir joué à un jeu vidéo au moins une fois au cours des six derniers mois

### 3. Enjeux statistiques de la plateforme Steam

L'analyse des données de Steam permet d'aborder plusieurs problématiques statistiques, telles que :

Prévision du succès d'un jeu : Quels facteurs influencent le nombre de joueurs et les ventes d'un jeu ?

Effet des promotions et des soldes : Les réductions ont-elles un impact durable sur la popularité des jeux ?

Analyse des avis et tendances : Peut-on détecter des corrélations entre les notes des joueurs et la longévité d'un jeu sur la plateforme ?

Segmentation des utilisateurs : Quels profils de joueurs émergent en fonction de leurs habitudes de consommation ?

L'étude des données Steam constitue donc un bon cas d'application pour les modèles statistiques, au sein d'un marché extrêmement dynamique et bien souvent médiatisé ou les facteurs de réussite d'un jeu sont étudiés attentivement par tous les développeurs avec une tendances significatives à la reproduction en masse de ces facteurs dans le but de meilleurs ventes (augmentation du temps de vie du jeu, notion d'open world...).

## Présentation des données

**Source** : <https://huggingface.co/datasets/FronkonGames/steam-games-dataset>

La base de données utilisée dans notre étude provient du jeu de données FronkonGames Steam Games Dataset, hébergé sur Hugging Face. Elle contient **83 600 observations**, chaque ligne représentant un jeu différent disponible sur la plateforme Steam.

### Fiabilité et origine des données

Les données ont été scrappées/récoltées par Martin Bustos, un créateur espagnol de jeux indépendants, mais aussi un « asset store publisher ». Il développe des «extensions/assets » pour Unity, un outil de développement de jeux vidéo puissant utilisé par une large partie des développeurs, qui constitue comme Steam, un pilier de l'industrie du jeu vidéo.

Pour s'aider dans ses activités et être plus pertinent, il a créé un script Python qui lui permet de récupérer des données directement accessibles depuis l'API Steam.

## « Qu'est-ce que l'API Steam ? »

L'API Steam est un service fourni par **Valve**, la société derrière la plateforme Steam. Elle permet aux développeurs d'accéder de manière sécurisée et fiable aux données de Steam, comme les informations sur les jeux, les joueurs, ou les succès (=accomplissement ou action particulière dans le jeu récompensée par un badge virtuel). Valve garantit l'intégrité et la fiabilité des données grâce à des mécanismes de sécurité robustes, comme l'authentification par clé API et le chiffrement des échanges.

Bien que Hugging Face ne soit pas une source gouvernementale ou officielle, nous avons accès au code source qui a permis l'extraction de ces données publiques, qui sont donc **authentiques et reproductibles**, en plus d'avoir directement servi Martin Bustos lui-même.

## Présentation des variables

### 1. Variable à expliquer

Nous avons choisi comme variable à expliquer (**Y**) :

**Average.playtime.forever**: qui mesure le nombre moyen de minutes passées sur un jeu par ses joueurs.

Ce choix repose sur l'hypothèse que la durée de jeu moyenne des joueurs est un bon indicateur de **l'intérêt et de l'engagement des joueurs** pour un titre donné.

Cette variable représente également un bon indicateur des revenus dégagés par un jeu, puisque le prix de ceux-ci ne s'arrête que rarement au prix d'achat (étant parfois même gratuit). En effet, la grande majorité des jeux disposent aujourd'hui de **contenu additionnel payant** (DLC, personnalisation), un joueur qui dispose d'un grand intérêt pour le jeu en termes de temps, sera ainsi **plus enclin à consommer** ces contenus.

L'importance économique de ces contenus additionnels étant non négligeable, par exemple le directeur de *Take-Two Interactive* (distributeur américain de jeux vidéo) a déclaré que 58 % de leurs revenus générés provenaient des DLC et des microtransactions, **surpassant ainsi la vente des jeux en tant que tel**.

### 2. Variables explicatives

Afin d'expliquer **Average.playtime.forever**, nous sélectionnons plusieurs variables quantitatives susceptibles d'influencer cette durée :

**Peak.CCU** (nombre maximum de joueurs simultanés) : Le nombre de joueurs est-il corrélé aux temps de jeu ? **Effet de volume ?**

**Required\_Age** (âge requis) : Les jeux destinés à un public plus restreint engendrent-ils un comportement différent ? ***Effet de ciblage ?***

**Price** (prix du jeu en Dollars) : Les jeux plus chers sont-ils plus engageants ?

***Effet de justification des coûts ?***

**PositiveReviews** (nombre d'avis positifs) : Un jeu bien noté est-il plus susceptible d'être joué en moyenne plus longtemps ? ***Effet de halo ?***

**NegativeReviews** (nombre d'avis négatifs) : Un jeu mal noté est-il susceptible d'être joué en moyenne moins longtemps ? ***Effet de rejet ?***

**Recommendations** (nombre de recommandations) : Il s'agit des jeux recommandés aux joueurs par l'algorithme de Steam, est-ce que ces recommandations influencent le comportement des joueurs ? ***Effet de recommandation sociale ?***

### 3. Population mère et échantillon

**Population mère** : l'ensemble des jeux publiés sur Steam en janvier 2024.

**Échantillon étudié** : Nous ne garderons dans la base de données que les jeux avec comme conditions :

**Peak.CCU** > 0 ; **PositiveReviews** > 0

Qui se traduit comme les observations ayant eu au moins 1 Joueur et 1 Avis Positif sur toutes périodes confondus.

## Exploration initiale des données

Avant de construire un modèle de régression, il est essentiel d'examiner la distribution des variables utilisées. Une analyse préliminaire permet d'identifier d'éventuelles asymétries et d'adapter le traitement des données en conséquence.

### Distribution des variables

Les variables quantitatives sélectionnées dans notre modèle (**peakccu**, **avg\_playtime**, **required\_age**, **price**, **positivereviews**, **negativereviews**, **recommendations**) présentent des écarts importants entre les jeux les plus populaires et les moins populaires. Cette inégalité se traduit par une distribution fortement asymétrique, avec une concentration de nombreux jeux ayant de faibles valeurs et quelques titres très populaires atteignant des valeurs extrêmes.

### Par exemple :

La majorité des jeux ont peu d'avis, tandis que quelques titres phares en cumulent des centaines de milliers.

Les prix varient de **0€** à plus de **100€**, avec une forte concentration dans la fourchette de 20 à 60€.

Le nombre maximal de joueurs simultanés (**peakccu**) est très élevé pour certains jeux, mais proche de zéro pour la majorité.

Il y a probablement différentes catégories de joueurs au **comportement et aux critères différents**, qu'il faudrait tenter d'identifier. Dans ce but on pourrait compléter l'analyse par l'ajout de **variables qualitatives**:

On pourrait ainsi se poser des questions autour du public visé, du genre de jeu, ou encore des modalités du jeu (seul ou multijoueur, en ligne ou hors ligne).

Mais avant leur ajout, une **transformation de nos données quantitatives** sera probablement nécessaire pour les rendre pertinentes dans un modèle linéaire

## Transformation des données

Face à ces déséquilibres, nous allons recourir à une transformation logarithmique en base 10 qui permettra de :

- **Réduire l'effet des valeurs extrêmes**, rendant les distributions plus normales.
- **Améliorer l'interprétabilité**, les variations deviennent plus lisibles, notamment pour les variables financières et les avis.
- **Stabiliser la variance**, facilitant l'ajustement d'un modèle linéaire fiable.

## Analyse des corrélations:

*Avant d'analyser les corrélations entre des variables quantitatives 2 à 2, il convient d'effectuer des tests de normalité sur nos variables transformés.*

Le test de normalité indique que **nos données ne suivent pas une distribution normale**. Nous utilisons donc le coefficient de corrélation de Spearman pour évaluer les relations entre les variables.

La variable "**peak.CCU**" présente la **corrélation la plus forte** avec la variable endogène (**0,5268**) illustrant qu'une augmentation du pic du nombre de joueurs connectés simultanément pourrait logiquement être associée à une augmentation du temps moyen de jeu et ainsi confirmer un certain effet de volume.

Les variables "**positives**", "**negatives**" et "**recommandations**" montrent également des **corrélations significatives**, comprises entre **0,41** et **0,45**. Cependant, ces dernières



sont **fortement corrélées entre elles** (supérieures à **0,8**), ce qui pourrait indiquer un **risque de multicollinéarité** perturbant nos modèles.

En réaction à la corrélation importante entre le nombre d'avis positifs et négatifs, nous allons les transformer en une variable unique en suivant les évaluations Steam.

## Création de la variable rating:

Puisque l'on constate que les variables "positive" et "négative" représentant respectivement les avis positifs et négatifs sont fortement corrélées, nous les regroupons en une seule variable qualitative que l'on nomme "rating" représentant l'avis global des joueurs divisé en neuf tranches allant de overwhelmingly negative à overwhelmingly positive.



illustration :

Titre : FARCRY 4

Avis positifs 👍: 35175

Avis négatifs 👎: 7255

rating : **Very positive**

## Élaboration d'un modèle:

### Modèle de base:

*Average.playtime.forever ~ Peak.CCU + Positive + Negative + Recommendations + Price + Required.age + Estimated.owners*

Nous commençons par un modèle naïf, incluant toutes les variables que nous avons supposé pertinentes pour la variable que nous souhaitons expliquer.

Notre premier constat est sur le coefficient de détermination, c'est-à-dire à quel point le modèle explique **la variance de la variable à expliquer, qui est à peine à 7%**. Ce n'est pas nécessairement un résultat choquant, en effet, notre exploration de données préliminaires montrait déjà des signes que nos données n'allaient pas suivre une relation linéaire (d'où le choix de visualiser certaines données sous transformation logarithmique).

Nous décidons de continuer l'analyse en vérifiant certaines hypothèses clés du modèle linéaire.

Premièrement en termes de linéarité des résidus, l'on observe en lieu et place d'une répartition uniforme sans motifs apparents entre les valeurs calculées par le modèle et



l'erreur associée, **une concentration dominante de valeurs faibles** avec quelques valeurs anormalement hautes. Cela correspond aux observations préliminaires: beaucoup de valeurs faibles mais quelques observations aux valeurs anormalement hautes.

On observe que les résidus sont très dispersés pour les faibles valeurs ajustées, tandis qu'ils sont plus regroupés pour les valeurs élevées. Cette forme en entonnoir inversé indique que **l'erreur n'est pas constante**, ce qui remet en question l'hypothèse d'homoscédasticité.

La **normalité des erreurs n'est également pas respectée**, en particulier aux extrémités de la distribution. On observe une asymétrie marquée, avec une concentration des résidus d'un côté et une dispersion plus importante de l'autre.

De plus, le modèle recense **plusieurs valeurs aberrantes** influentes, perturbant considérablement l'ajustement du modèle.

En dépit de ces limites, on peut toutefois souligner un point positif, **l'indépendance des erreurs** semble globalement respectée.

De plus, l'analyse du VIF **ne met pas en évidence de multicollinéarité** marquée entre les différentes variables. On observe toutefois une corrélation plus prononcée entre les variables 'positives' et 'recommandations' qu'entre 'positives' et 'négatives', ce qui nuance notre hypothèse initiale. Cela suggère que la variable 'recommandations', régie par l'algorithme de Steam, pourrait s'appuyer en grande partie sur les avis positifs des joueurs.

**Naturellement nous rejetons ce modèle**, offrant un pouvoir prédictif trop bas en plus de ne pas respecter certaines hypothèses essentielles du modèle linéaire.

*Nous allons donc réaliser une transformation logarithmique de la variable dépendante (et de certaines variables explicatives) afin de réduire l'influence des valeurs extrêmes et améliorer la qualité globale de l'ajustement.*

## Transformation logarithmique:

Nous appliquons donc un  $\log_{10}$  aux variables présentant une forte asymétrie, à l'exception de **"Required.age"**, qui est déjà une échelle discrète limitée (0, 12, 16, 18 ans).

A la suite de cette transformation, l'on remarque que le pouvoir explicatif du modèle est bien plus élevé avec **un coefficient de détermination avoisinant les 30%**, un résultat bien meilleur que celui de notre premier modèle mais toujours assez médiocre.

Ce second modèle révèle une distribution globalement satisfaisante des résidus. **L'hypothèse d'homoscédasticité semble mieux respectée**, avec une dispersion relativement constante des résidus tout au long des valeurs ajustées. Aucun motif clair de forme en éventail n'est observable, ce qui indique que la variance des erreurs reste stable. Cela suggère que la transformation logarithmique a permis de corriger en partie les problèmes de variance observés dans le modèle initial.

En revanche, quelques observations atypiques conservent une forte influence sur le modèle et mériteraient un traitement spécifique (suppression ou robustification du modèle).

Par ailleurs, le test de Durbin-Watson donne une statistique proche de 2 ( $DW = 1.94$ ), avec une p-value significative (0.0013), ce qui suggère une très faible mais significative **autocorrélation positive des erreurs de premier ordre**, là où notre précédent modèle n'en indiquait pas.

Cette autocorrélation est d'autant plus confirmée par l'analyse de l'ACF, où de nombreuses barres se situent nettement au-dessus de la zone de confiance. Cela met en évidence la présence d'une structure de dépendance entre les résidus et enfreint ainsi l'hypothèse d'autocorrélations de ceux-ci. Ce nouveau modèle apporte donc des problèmes d'autocorrélations non négligeables.

*L'on observe ainsi globalement une amélioration du modèle mais celui-ci reste tout de même assez médiocre, en plus d'apporter des problèmes d'autocorrélations des erreurs. Nous allons donc réaliser une approche plus radicale en supprimant les données jugées nuisibles à l'établissement de celui-ci dans l'idée d'obtenir un modèle plus robuste.*

## Un nettoyage accru des données:

Nous procédons ainsi à la suppression des données présentant des résidus très conséquents et/ou une influence trop importante sur le modèle de part un comportement atypiques.

### **1263 points sont alors supprimés.**

Malgré tout, les résultats obtenus ne diffèrent guère de ceux du modèle précédent, avec un coefficient de détermination légèrement supérieur à 30 %.

*En somme, bien que quelques améliorations aient été apportées, notre modèle conserve des performances très faibles et **n'est pas capable en l'état d'expliquer le temps de jeu moyen des joueurs.***

## L'existence de sous-populations ? :

Face à la faiblesse persistante des performances du modèle, malgré les transformations et le nettoyage des données, une hypothèse alternative mérite d'être envisagée celle de la **présence de sous-populations** au comportement spécifique au sein de notre base.

En effet, la plateforme Steam regroupe une grande variété de jeux, allant de petits jeux indépendants à très faible durée de vie jusqu'aux blockbusters à très fort taux de rétention. Il est donc plausible que le comportement moyen des joueurs varie considérablement selon le type de jeu, sa notoriété ou encore son modèle économique (free-to-play, premium, abonnement, etc...).

Dans ce cadre, nous faisons notamment l'hypothèse que ces sous-groupes pourraient être **structurés autour de l'éditeur du jeu**. En effet, chaque éditeur possède ses propres caractéristiques ; **public ciblé, style de jeu développé, moyens de production, budget marketing, réputation etc...**

Par exemple, un grand éditeur comme Ubisoft ou Valve aura tendance à produire des titres à gros budget, souvent conçus dans sa structure pour favoriser une forte rétention et une longue durée de vie, le but primaire étant pour ses structures de grande envergure de **satisfaire les actionnaires** finançant le projet. À l'inverse, les petits studios indépendants publient souvent des jeux de niche, à durée de vie plus courte ou au gameplay plus expérimental dans l'objectif de **se faire connaître**, se construire une communauté fidèle.

Cette hétérogénéité des profils éditoriaux peut expliquer en partie l'échec de notre modèle global à bien prédire les comportements moyens.

Cela ouvre ainsi la voie à une **segmentation préalable** des jeux selon leur éditeur, permettant d'identifier des dynamiques propres à chaque sous-population. Une telle approche pourrait révéler des relations plus nettes entre les variables explicatives et le temps de jeu, masquées jusqu'ici par l'agrégation.



Dans cette optique, nous allons ainsi étudier notre hypothèse auprès de l'éditeur français le plus représenté dans notre base de données : Ubisoft.

La quantité importante de jeux publiés par Ubisoft (100+) sur la plateforme permet en effet de disposer d'un échantillon suffisamment large et varié pour mener une analyse pertinente.



## Solutions possibles: la segmentation par éditeurs

L'on peut observer à la suite de cette approche que notre modèle préalablement segmenté possède un coefficient de détermination supérieur à 60% offrant à celui-ci **un pouvoir explicatif bien plus supérieur** que nos précédents modèles.

De plus, en termes d'autocorrélation des erreurs, l'ACF semble indiquer qu'il n'y a pas de corrélation. Cependant, le test de Durbin-Watson fournit une statistique de 1,74, ce qui suggère une faible corrélation positive des erreurs, ainsi il est plausible que notre modèle possède bien une corrélation mais uniquement de premier ordre.

Malgré tout les résidus ne semblent toujours pas être linéaires, se traduisant graphiquement par une forme incurvée, de même l'homoscédasticité de ceux-ci est compromise, la variance des résidus étant plus grande pour les petites valeurs ajustées et semble se stabiliser ensuite.

Une transformation logarithmique se révélant après test peu utile dans ce cas.

*En somme, notre approche segmentée offre donc un pouvoir explicatif de temps de jeu moyen bien plus intéressant mais des problèmes persistent concernant les hypothèses sur les résidus rendant le modèle peu robuste.*

## Optimisation des modèles :

Par la suite nous allons donc conserver ce modèle bien que limités, et tester trois critères d'évaluation pour l'optimisation de celui-ci, **AIC, BIC et le test de Fischer**, en commençant par une sélection via *glmulti*, puis en poursuivant avec une méthode *step by step*

Le constat établi à partir de ces critères semble assez clair, puisqu'ils convergent tous vers le même modèle comme étant optimal :

$$\text{Average.playtime.forever} \sim \text{Negative} + \text{Recommendations} + \text{Price} + \text{Required.age}$$

Les variables **"Price" et "Negative"** semblent ainsi être les plus influentes. La seule différence observée entre les différents critères est l'absence de la variable "Recommendations" dans le modèle basé sur le BIC.

Ces résultats sont étonnants au regard des corrélations observées entre les variables précédemment, on s'attendait notamment à ce que les variables "Positive" et "Peak.CCU" soient priorisées. Cependant il reste **possible que ces conclusions soient spécifiques à Ubisoft** et ne se répètent pas pour d'autres éditeurs.

Enfin, en termes de pouvoir explicatif, on remarque que ces modèles plus simples conservent **un  $R^2$  ajusté quasiment identique** à celui du modèle complet pour Ubisoft présenté précédemment.

## D'autres approches:

Dans une optique d'approfondissement de l'analyse nous avons mis en place des approches supplémentaires aux modèles linéaires, dans l'espoir de construire un modèle plus performant ou de pousser notre analyse des données .

### Modèles polynomiaux :

Afin de mieux capturer les **relations non linéaires** entre certaines variables explicatives et le temps de jeu moyen, des **modèles polynomiaux** ont été utilisés. Plus précisément, des régressions polynomiales de degré 1 à 3 ont été testées individuellement pour chaque variable, afin de visualiser si une modélisation plus souple permettrait d'améliorer la qualité de l'ajustement.

Dans la majorité des cas, les modèles polynomiaux de degré 2 offrent une **légère amélioration** par rapport au modèle linéaire, comme en témoignent les baisses modérées de l'AIC et les hausses marginales du  $R^2$  ajusté. Toutefois, ces gains restent **extrêmement limités**, et les performances globales des modèles demeurent très faibles. Il semble donc peu pertinent d'explorer davantage ce type de modèle, compte tenu du peu d'amélioration obtenue sur les modèles simples à 1 variable.

### Classification/Modèle logistique :

Dans cette seconde approche, nous avons choisi de modifier la variable à prédire Y pour s'intéresser au nombre de joueurs estimés (Estimated . owners). Nous utilisons ici un modèle logistique multinomial, ce qui nous fait rester sur des relations linéaires entre les variables explicatives et la variable cible.

*Décomposition de la variable Estimated.Owners : [0-20k], [20k-50k], [50k-100k], [100k-200k], [200k-500k], [500k-1M], [1M-2M], [2M-5M], [5M-10M], [10M-20M], [20M-50M], [50M-100M] et [100M-200M]*

Ainsi, il est important de noter que la distribution des classes de "Estimated.owners" est loin d'être uniforme (peu de jeux peuvent prétendre à plus de 500k ventes), ce qui peut entraîner des résultats déséquilibrés dus à la sous représentation de certaines classes.

Les variables liées à l'engagement des joueurs comme "Recommendations", "Positive" et "Average.playtime.forever" ressortent comme très significatives dans la majorité des classes. Toutefois, on observe une tendance intéressante : **la significativité du temps de jeu ("Average.playtime.forever") diminue pour les classes les plus élevées.**

Par exemple, dans les tranches supérieures comme 10M-20M, 20M-50M ou 50M-100M, cette variable devient non significative, ce qui indique que pour les jeux extrêmement populaires, la durée moyenne du temps de jeu n'est plus un bon indicateur du nombre de possesseurs. Cela pourrait être dû à la diversité des profils de joueurs ou à la présence de jeux très populaires mais peu chronophages.

La variable "Price", quant à elle, est significative dans la quasi-totalité des classes, ce qui montre **un lien étroit entre le prix du jeu et sa diffusion**, probablement en raison de stratégies commerciales différenciées selon la popularité attendue.

Du côté de la multicolinéarité, les VIF obtenus ne sont pas préoccupants, ce qui suggère une relative indépendance entre les variables explicatives. Le modèle atteint un **taux de bonnes prédictions de 44 %**, ce qui, bien que modeste, constitue une amélioration significative par rapport à un modèle trivial, qui plafonne à 17 %.

Enfin, il est à souligner que les variables ont été transformées par logarithme et standardisées, ce qui a permis d'atténuer les effets de dispersion et d'échelle.

En somme, dans un contexte où les relations sont loin d'être linéaires et où certaines classes sont largement sous-représentées, obtenir une précision de 44 % reste un **résultat globalement encourageant** sur la qualité de nos prédicteurs.

## Conclusion :

L'ensemble des analyses menées met en évidence la complexité des comportements liés au temps de jeu et à la diffusion des jeux sur Steam. Si nos premiers modèles linéaires globaux se sont révélés peu performants, avec un pouvoir explicatif très limité, des approches plus ciblées – notamment par éditeur – ont permis d'obtenir des résultats plus satisfaisants, atteignant jusqu'à 60 % de variance expliquée résultant de la présence d'un frottement possible de sous-populations spécifique au sein de notre base.

En parallèle, notre tentative de classification du nombre de joueurs via un modèle logistique multinomial a montré une précision non négligeable de 44 %, malgré une distribution très déséquilibrée des classes et ce problème de non linéarité que nous n'avons pas réussi à compenser via la transformation logarithmique.

## Limites et perspectives

Plusieurs limites restent à souligner. D'une part, les hypothèses fondamentales des modèles linéaires sont régulièrement mises à mal (autocorrélation, hétéroscédasticité, non-normalité des résidus), ce qui fragilise leur fiabilité. D'autre part, la très forte hétérogénéité des jeux en termes de type, d'audience, de stratégie commerciale tend à brouiller les relations globales entre variables.

Une segmentation plus fine, non seulement par éditeur, mais aussi par genre, modèle économique, pourrait permettre d'identifier des sous-dynamiques plus homogènes. On pourrait bénéficier également d'analyses démographiques sur les joueurs, de données plus spécifiques sur les éditeurs ou du traitement des données temporelles. Enfin, le recours à des modèles non linéaires (forêts aléatoires, réseaux de neurones, etc.) ou à des approches bayésiennes constituerait une piste intéressante pour capturer plus efficacement la richesse et la complexité du marché du jeu vidéo.

## Sources:

Header: [https://cdn.fastly.steamstatic.com/store/home/store\\_home\\_share.jpg](https://cdn.fastly.steamstatic.com/store/home/store_home_share.jpg)

Statistiques sur Steam:

<https://store.steampowered.com/charts>

<https://steamcharts.com/>

<https://steamdb.info/charts/>

Jeu de données: <https://huggingface.co/datasets/FronkonGames/steam-games-dataset>

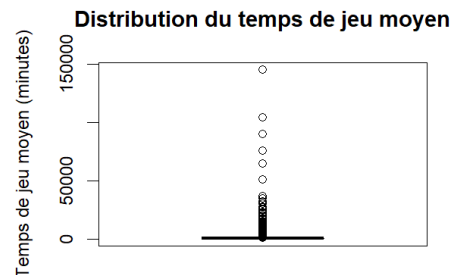


GitHub de Martin Bustos: <https://github.com/FronkonGames>

## Annexe:

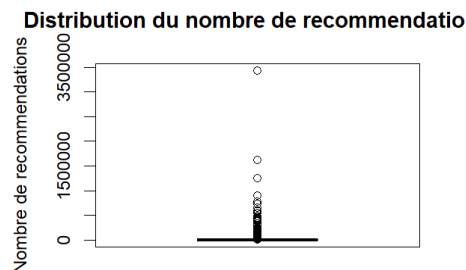
### Tests univariés

#### Distribution du temps de jeu moyen



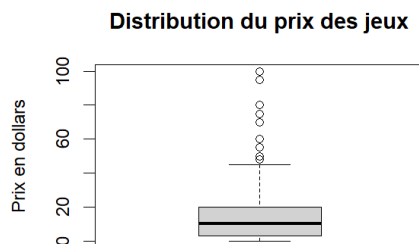
Le temps de jeu moyen varie considérablement, avec des valeurs allant de 0 à plusieurs centaines de milliers de minutes. On remarque une distribution asymétrique du temps de jeu moyen avec une médiane relativement basse, ce qui indique que la plupart des jeux ont un temps de jeu moyen modéré, tandis que quelques jeux ont des temps de jeu extrêmement élevés.

#### Distribution du nombre de recommandations



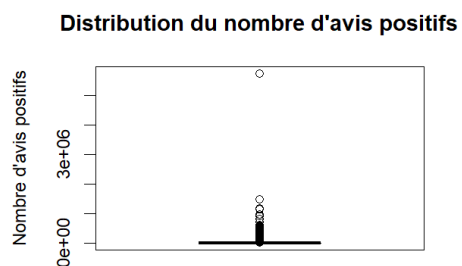
La majeure partie des jeux possède un nombre de recommandations relativement faibles, concentrées autour de la médiane. Cependant, la présence de nombreux valeurs extrêmes indique que certains jeux ont reçu un nombre de recommandations extrêmement élevé par rapport aux autres. Cette distribution indique que seuls certains jeux particulièrement populaires dominent le classement.

### Distribution du prix



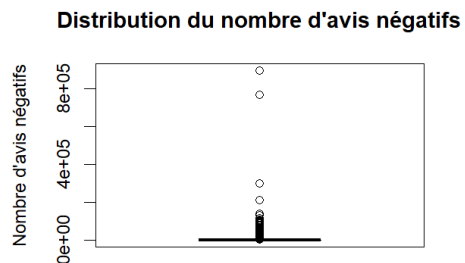
Dans ce boxplot qui illustre la distribution des prix des jeux en dollars on observe que la majorité des jeux ont un prix situé dans une fourchette relativement basse, et proches les uns des autres. La grande majorité des jeux ont ainsi un prix entre 0 et 50€.

### Distribution du nombre d'avis positifs



Le nombre d'avis positifs varie considérablement, avec des valeurs allant de 0 à plusieurs millions. Certains jeux ont un nombre d'avis positifs bien supérieur à la majorité des autres d'où cet écart entre les valeurs. Cela pourrait refléter des différences significatives dans la popularité ou la satisfaction des joueurs selon les jeux.

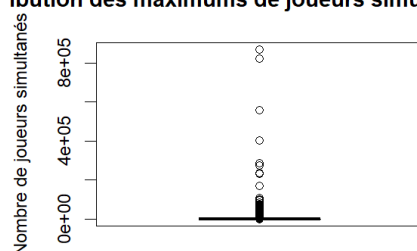
### Distribution du nombre d'avis négatifs



Tout comme la distribution du nombre d'avis positifs, le nombre d'avis négatifs varie considérablement, avec des valeurs allant de 0 à plusieurs millions. Certains jeux ont un nombre d'avis négatifs bien supérieur à la majorité des autres d'où cet écart entre les valeurs. Cela pourrait refléter des différences significatives dans la popularité ou la satisfaction des joueurs selon les jeux.

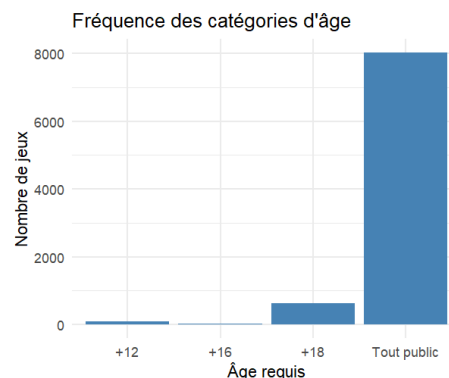
### Distribution du nombre maximum de joueurs simultanés

**Distribution des maximums de joueurs simultanés**



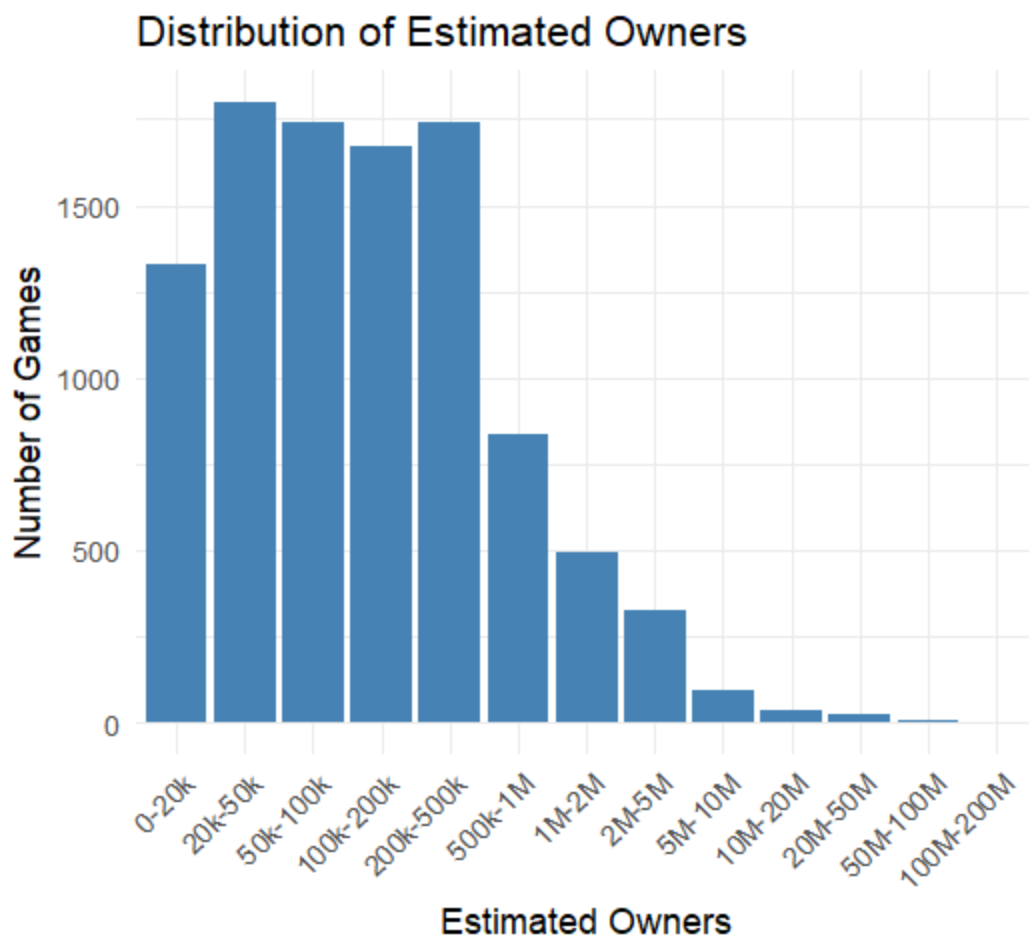
Cette boîte à moustache met en évidence une distribution asymétrique des maximums de joueurs simultanés, avec une concentration de jeux ayant des pics modérés et quelques jeux ayant des pics extrêmement élevés. La plupart des pics de connexion se produisent dans une plage d'horaire spécifique. Les points extrêmes représentent des moments de la journée où les pics de connexion sont inhabituels.

### Distribution des catégories d'âges



Les jeux qui sont sans limite d'âge requis dominent largement le marché avec une offre importante de près de 8000 types de jeux. On observe ainsi une sous population potentiellement non négligeable de jeux matures.

### Distribution du nombre de propriétaires estimé

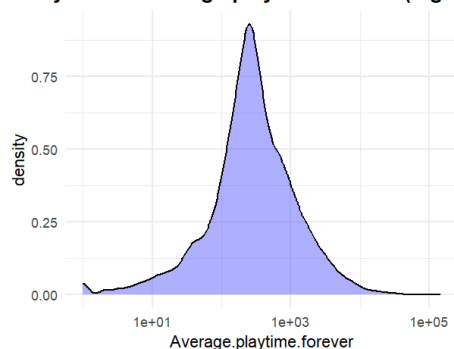


La majorité des jeux possède entre 0 et 500k acheteurs, et on observe un rapide déclin du nombre de jeux au-dessus de 500k acheteurs. Quelques jeux ont une popularité extrême.

## Tests univariés sous transformations logarithmique

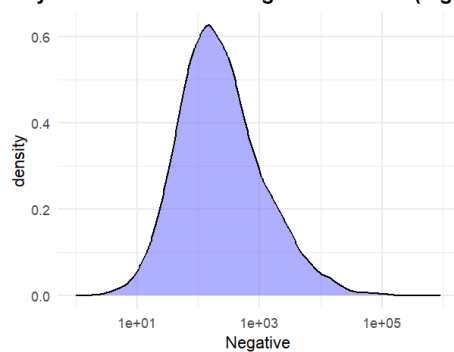
### Distribution du temps de jeu moyen sous transformation logarithmique

ensity Plot of Average playtime forever (log10)



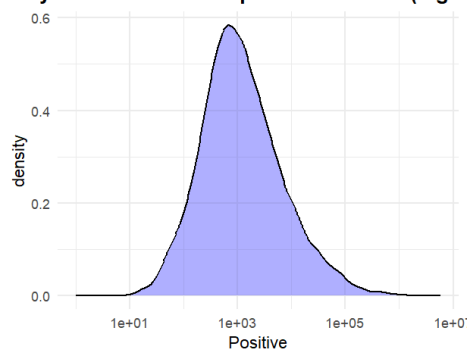
### Distribution du nombre d'avis négatifs sous transformation logarithmique

ensity Plot of number of negative reviews (log10)

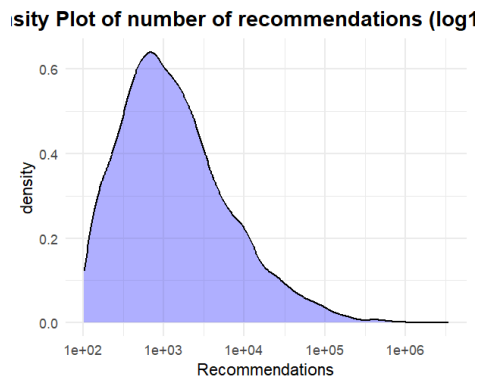


### Distribution du nombre d'avis positifs sous transformation logarithmique

ensity Plot of number of positive reviews (log10)

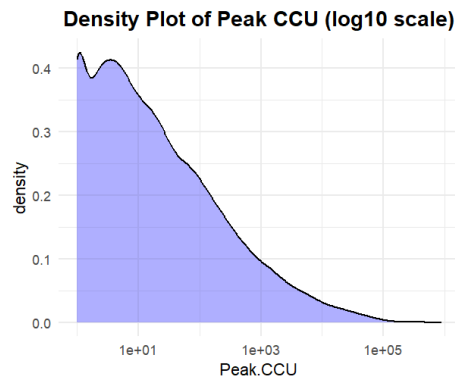


### Distribution du nombre de recommandations sous transformation logarithmique



Les transformations logarithmiques du dessus montrent des graphiques plus faciles à comprendre, plus proches d'une distribution normale, mais il ne faut pas oublier que les valeurs sont en réalité très concentrées dans les valeurs faibles avec des extrêmes très hauts.

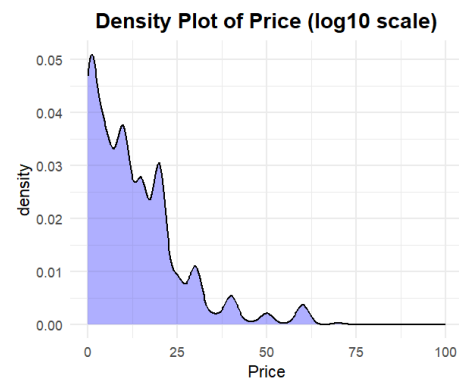
### Distribution du nombre maximum de joueurs simultanés sous transformation logarithmique



Malgré le logarithme on a une concentration dominante dans les valeurs faibles avec un rapide décroissance pour les valeurs plus hautes. Ce qui indique une écrasante majorité de jeux avec un pic de joueurs bas.

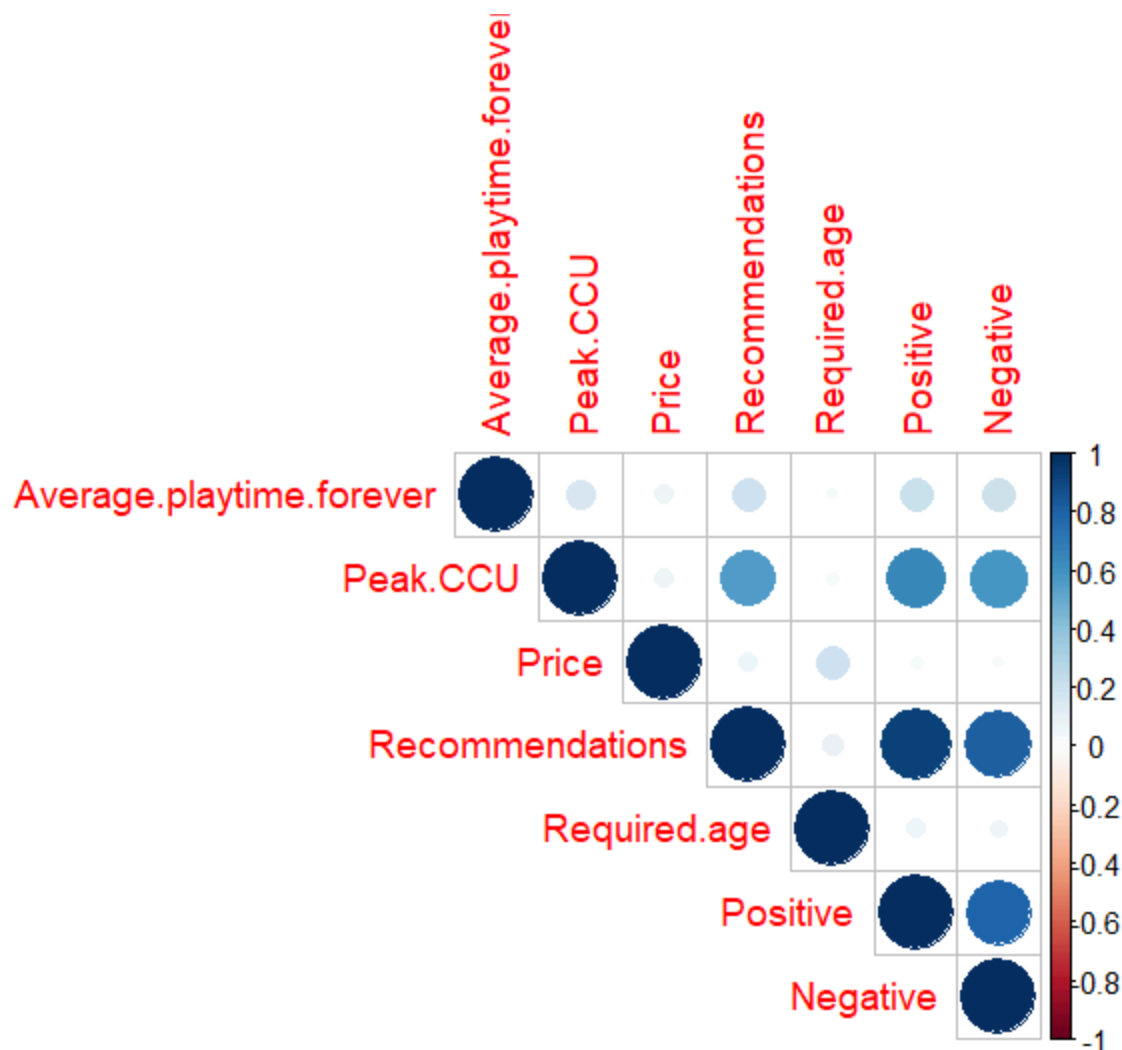


### Distribution du prix sous transformation logarithmique



En plus de la distribution asymétriques, il est intéressant de noter les différents pics de prix, indiquant des concentrations particulières autour de certains prix spécifiques, qui peut être issu d'une stratégie marketing globale, ou intrinsèquement lié aux standards des coûts de production d'un jeu-vidéo.

## Matrice de corrélation



Toutes les corrélations sont positives, bien que certaines sûrement négligeables, on fera attention au triplet Recommendations Positive et Negative qui ont l'air fortement corrélés, ce qui pourrait s'avérer gênant pour nos modèles. On retiendra également Peak.CCU qui a une corrélation positive non négligeable avec le triplet mentionné.

## Tableau évaluations Steam

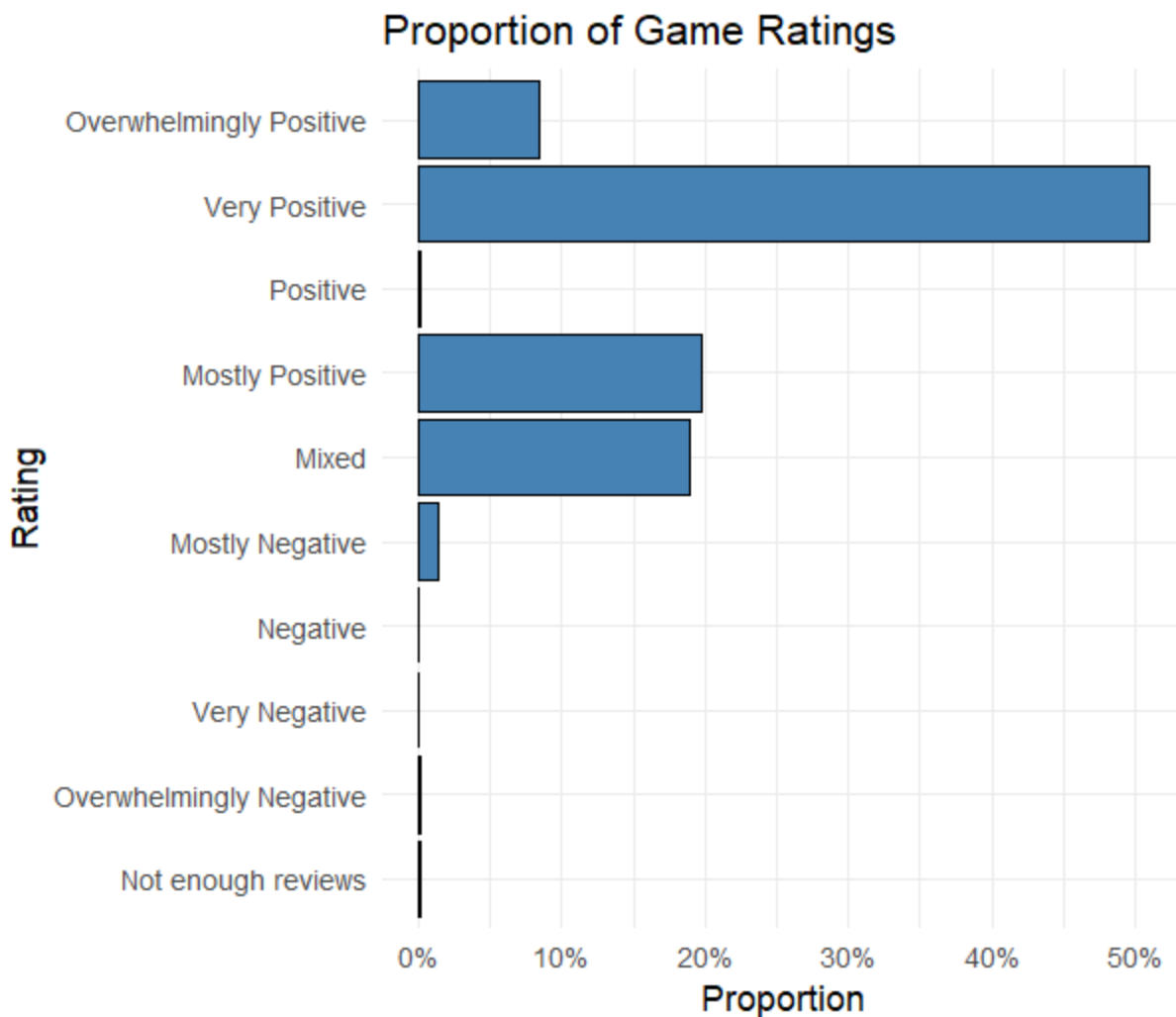
Positive %	10-49 reviews	50-499 reviews	500+ reviews
95%-100%	Positive 80%-100%	Very Positive 80%-100%	Overwhelmingly Positive
90%-94%			Very Positive 80%-94%
85%-89%			
80%-84%			
75%-79%	Mostly Positive 70%-79%		
70%-74%			
65%-69%	Mixed 40%-69%		
60%-64%			
55%-59%			
50%-54%			
45%-49%			
40%-44%			
35%-39%	Mostly Negative 20%-39%		
30%-34%			
25%-29%			
20%-24%			
15%-19%	Negative 0%-19%	Very Negative 0%-19%	Overwhelmingly Negative 0%-19%
10%-14%			
5%-9%			
0%-4%			

Seemingly how Steam ratings work - chart by @runevision Sep 2020

So in summary: to get Overwhelmingly Positive Steam reviews, you'll need at least 500 reviews and at least 95% of them have to be positive.

Les pages de la boutique Steam offrent un système d'évaluations basé sur le nombre d'avis positifs et négatifs. Et ce tableau montre les critères d'attributions. Nous allons créer une nouvelle variable rating dans nos données afin de refléter ces évaluations.

## Nouvelle variable qualitative Rating



Malgré la distribution d'avis négatifs observés auparavant, très peu de jeux se voient attribuer une évaluation négative. Il est probable que lorsqu'un jeu est mauvais, les joueurs ne s'embêtent pas à laisser un avis négatif et préfèrent l'ignorer.

Le peu de jeux évalué en Positive nous fait penser que lorsqu'un jeu commence à devenir populaire il dépasse les 50 avis, d'où le trou apparent dans l'histogramme.

## Tests bivariés

```
> ## Afficher tout les resultats des tests bivaries----
> print(lillie_table)
```

	p.value.lillie.tests
Average.playtime.forever	1.622262e-126
Peak.CCU	0.000000e+00
Price	0.000000e+00
Recommendations	0.000000e+00
Required.age	0.000000e+00
Positive	3.275849e-44
Negative	1.265541e-34
total_reviews	6.808082e-53
positive_ratio	0.000000e+00

```
> print(spearman_table)
```

Le test de lillie est un test de normalité, les faibles p-values nous indiquent qu'aucune des variables ne suit une loi normale. Rien d'étonnant avec les observations précédentes. Les graphiques sous transformations logarithmiques suggèrent que même sous la transformation logarithmique, les valeurs ne suivent toujours pas une loi normale.

```
> print(spearman_table)
```

	p.value.spearman	rho..Spearman.
Peak.CCU	0.000000e+00	0.50220517
Price	3.625868e-168	0.27006161
Recommendations	0.000000e+00	0.43615339
Required.age	1.762948e-36	0.12509412
Positive	0.000000e+00	0.45098029
Negative	0.000000e+00	0.41285270
total_reviews	0.000000e+00	0.45316951
positive_ratio	1.276234e-23	0.09948767

```
> print(kruskal_table)
```

Puisque les hypothèses de normalité ne sont pas respectées, il convient d'effectuer le test de Spearman, qui ne nécessitent pas la normalité, afin de mesurer la force et la direction d'une potentielle relation monotone entre les variables explicatives et Average.playtime.forever.

Peak.CCU :

p-value = 0, ce qui signifie que la corrélation est statistiquement significative.

Spearman's rho = 0.5022, ce qui montre une corrélation positive modérée entre Peak.CCU et Average.playtime.forever.

Price :

p-value = très faible (presque nulle), donc la corrélation est significative.

Spearman's rho = 0.2701, ce qui indique une corrélation positive faible entre Price et Average.playtime.forever.

Recommendations :

p-value = 0, donc significatif.

Spearman's rho = 0.4362, une corrélation positive modérée entre Recommendations et Average.playtime.forever.

Required.age :

p-value = très faible, donc la corrélation est significative.

Spearman's rho = 0.1251, une corrélation positive faible entre Required.age et Average.playtime.forever.

Positive :

p-value = 0, donc significatif.

Spearman's rho = 0.4510, une corrélation positive modérée entre Positive et Average.playtime.forever.

Negative :

p-value = 0, donc significatif.

Spearman's rho = 0.4129, une corrélation positive modérée entre Negative et Average.playtime.forever.

total\_reviews :

p-value = 0, donc significatif.

Spearman's rho = 0.4532, une corrélation positive modérée entre total\_reviews et Average.playtime.forever.

positive\_ratio :

p-value = très faible, donc significatif.

Spearman's rho = 0.0995, une corrélation positive faible entre positive\_ratio et Average.playtime.forever.

En résumé :

Il existe plusieurs variables qui ont des corrélations positives modérées avec Average.playtime.forever, telles que Peak.CCU, Recommendations, Positive, Negative, et total\_reviews.

Certaines variables ont des corrélations faibles, mais significatives, comme Price, Required.age, et positive\_ratio.

Les p-values proches de 0 indiquent que toutes ces relations sont statistiquement significatives.

```
> print(kruskal_table)
```

	Test	H_statistic	p_value
Kruskal-Wallis chi-squared	Estimated.owners	1721.6919	0.000000e+00
Kruskal-Wallis chi-squared1	rating	157.3566	2.628989e-29

```
> |
```

Interprétation des résultats :

Estimated.owners :

H\_statistic = 1721.6919

p\_value = 0 (c'est-à-dire que la p-value est extrêmement petite).

Interprétation : Il y a des différences significatives dans la distribution de la variable Estimated.owners entre les groupes. Cela signifie que les groupes basés sur Estimated.owners ont des distributions statistiquement différentes.

rating :

H\_statistic = 157.3566

p\_value = 2.628989e-29 (encore une p-value extrêmement petite).

Interprétation : Il y a également des différences significatives dans la distribution de la variable rating entre les groupes. Comme pour Estimated.owners, cela suggère que les groupes définis par rating ont des distributions différentes.

En résumé :

Les deux tests montrent qu'il y a des différences significatives dans les distributions des variables Estimated.owners et rating en fonction des groupes que tu as définis, ce qui signifie que ces variables sont influencées par la variable catégorielle associée.



## Premier modèle naïf

Average.playtime.forever en fonction de Peak.CCU, Positive, Negative, Recommendations, Price, Required.age et la variable qualitative Estimated.owners

```
> model <- create_lm(gamesc, Y, X, categories)
> summary(model)
```

Call:  
lm(formula = formula, data = dataset)

Residuals:

Min	1Q	Median	3Q	Max
-5881	-518	-329	-106	145089

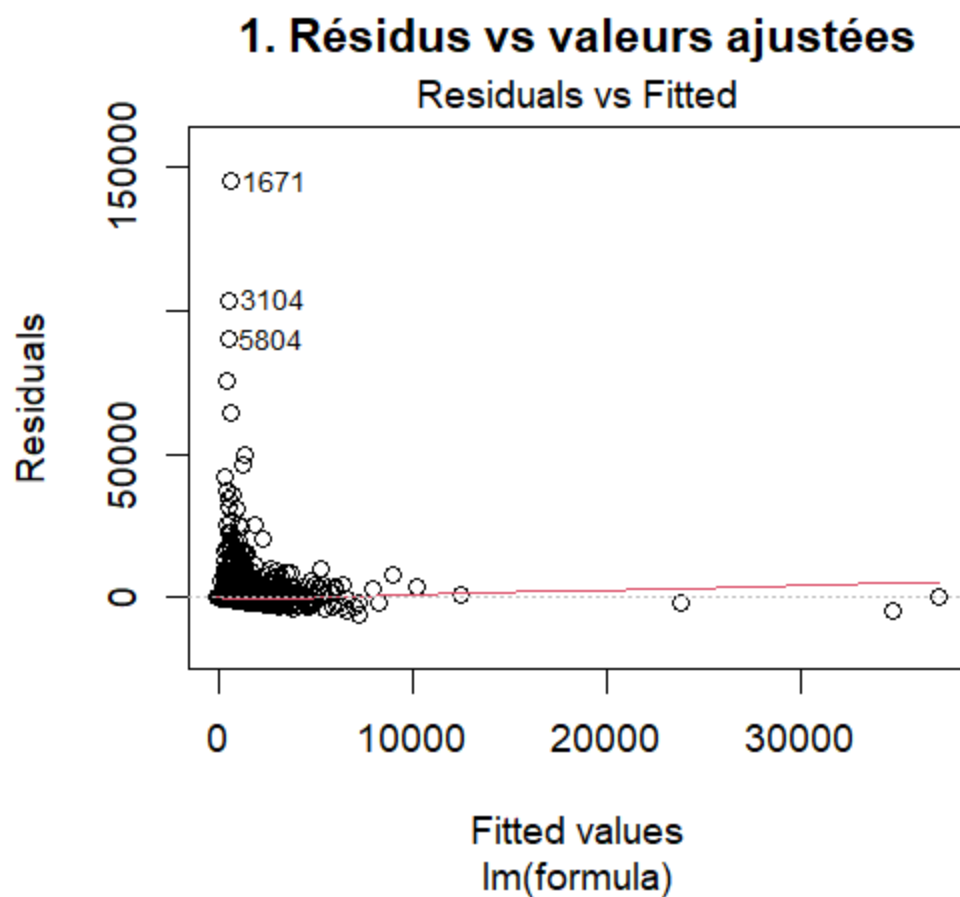
Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	2.436e+02	8.609e+01	2.830	0.00467	**
Peak.CCU	2.060e-04	2.582e-03	0.080	0.93642	
Positive	1.512e-03	1.471e-03	1.028	0.30392	
Negative	1.539e-02	5.062e-03	3.040	0.00237	**
Recommendations	3.158e-03	2.335e-03	1.353	0.17617	
Price	1.686e+01	2.339e+00	7.209	6.06e-13	***
Required.age	-1.800e+01	6.798e+00	-2.648	0.00812	**
Estimated.owners100k-200k	7.222e+01	1.064e+02	0.679	0.49724	
Estimated.owners1M-2M	7.714e+02	1.547e+02	4.987	6.23e-07	***
Estimated.owners10M-20M	2.006e+03	5.035e+02	3.984	6.83e-05	***
Estimated.owners100M-200M	2.990e+04	4.110e+03	7.275	3.73e-13	***
Estimated.owners20k-50k	1.053e+02	1.046e+02	1.007	0.31416	
Estimated.owners200k-500k	2.981e+02	1.056e+02	2.822	0.00478	**
Estimated.owners2M-5M	1.115e+03	1.841e+02	6.058	1.43e-09	***
Estimated.owners20M-50M	3.521e+03	7.002e+02	5.028	5.03e-07	***
Estimated.owners50k-100k	2.036e+02	1.054e+02	1.932	0.05334	.
Estimated.owners500k-1M	5.562e+02	1.284e+02	4.331	1.50e-05	***
Estimated.owners5M-10M	1.871e+03	3.177e+02	5.890	3.98e-09	***
Estimated.owners50M-100M	2.929e+03	2.236e+03	1.310	0.19024	

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

Residual standard error: 2888 on 10072 degrees of freedom  
Multiple R-squared: 0.06796, Adjusted R-squared: 0.06629  
F-statistic: 40.8 on 18 and 10072 DF, p-value: < 2.2e-16

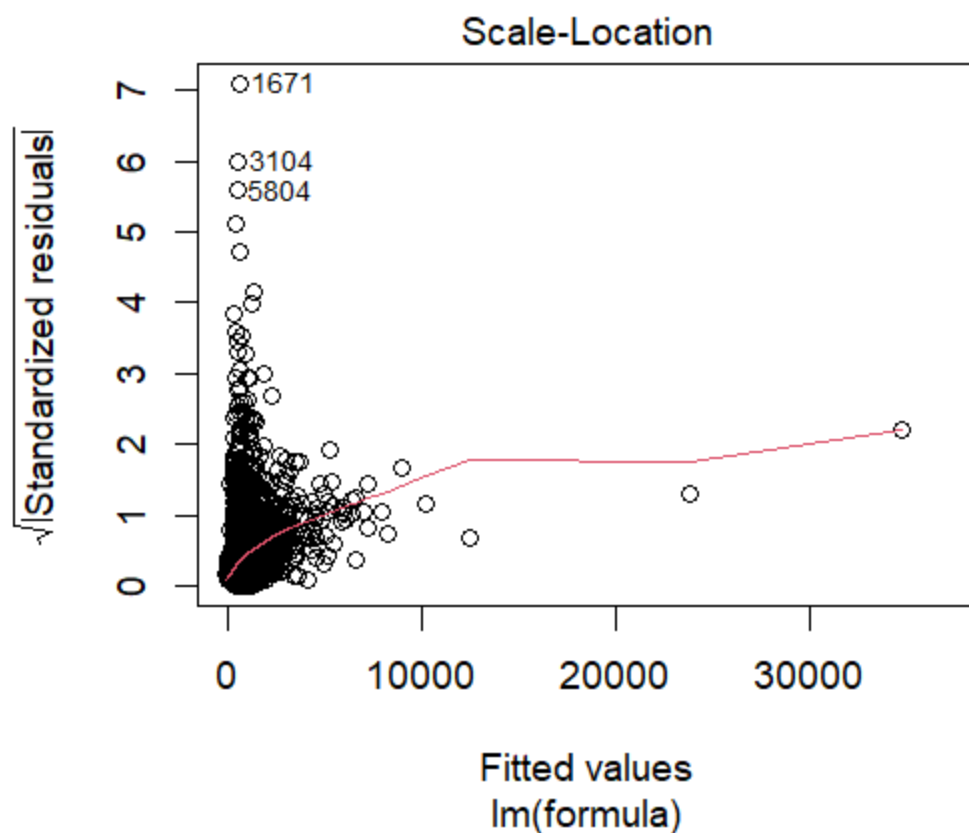
Bien que la p-value soit bonne, les coefficients de détermination  $R^2$  et  $R^2$  ajustés sont extrêmement bas. Malgré son pouvoir explicatif très bas nous allons tester plusieurs hypothèses pour comprendre la pauvre performance du modèle.



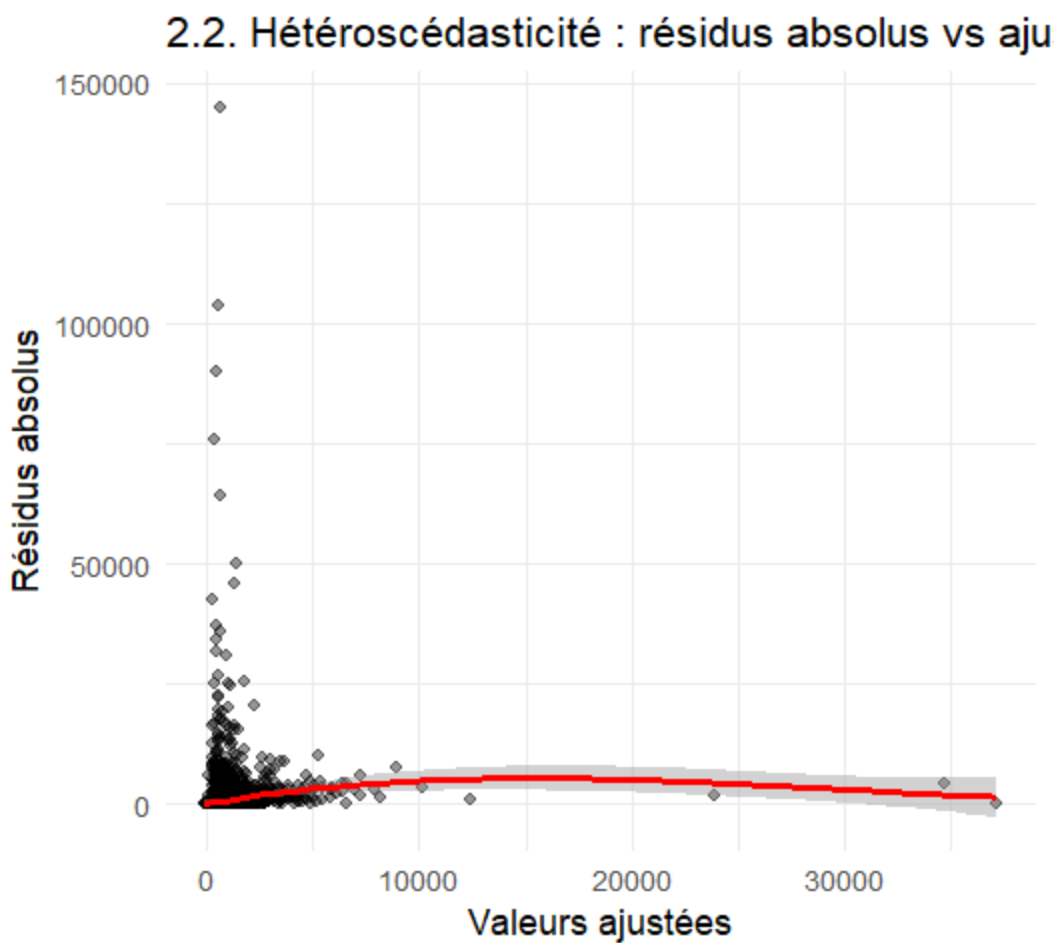
En cas de linéarité, le nuage de points doit être centré autour de 0 sans motif évident. Comme attendu, la linéarité est très mauvaise, voire non-existante.

Ici il y a une forte concentration de points à gauche, et verticalement les résidus ne sont pas bien répartis non plus.

## 2.1. Écarts à l'effet de levier



Au-delà de la forte concentration des points sur la gauche, la ligne rouge montre que l'erreur n'est en moyenne pas constante, remettant en cause l'homoscédasticité des erreurs.



Malgré la forme du nuage de points très irrégulière, la variance des erreurs semble constante.

```

> check_lm_hypotheses(model, gamesc)
Vérification des hypothèses pour le modèle : lm(formula = formula, data = dataset)

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

Test de Durbin-Watson (attendu ≈ 2) :

      Durbin-Watson test

data:  model
DW = 2.0012, p-value = 0.5243
alternative hypothesis: true autocorrelation is greater than 0

VIF (Variance Inflation Factor) :
      GVIF Df GVIF^(1/(2*Df))
Peak.CCU      1.978060  1      1.406435
Positive     12.850608  1      3.584774
Negative      5.376624  1      2.318755
Recommendations 15.405534  1      3.924988
Price         1.066061  1      1.032503
Required.age   1.113434  1      1.055194
Estimated.owners 5.698648 12      1.075203

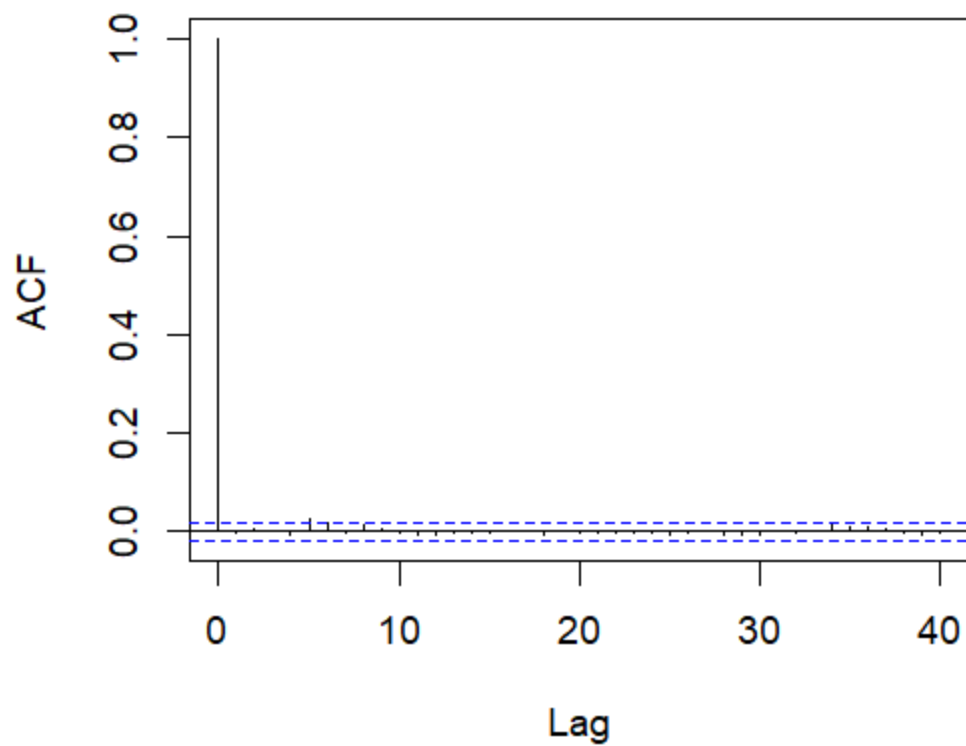
Variables avec VIF > 5 :
NULL

```

Le test de Durbin Watson retourne une valeur de 2 et une grande p-value, ce qui confirme qu'il n'y a pas d'auto corrélation des erreurs de premier ordre.

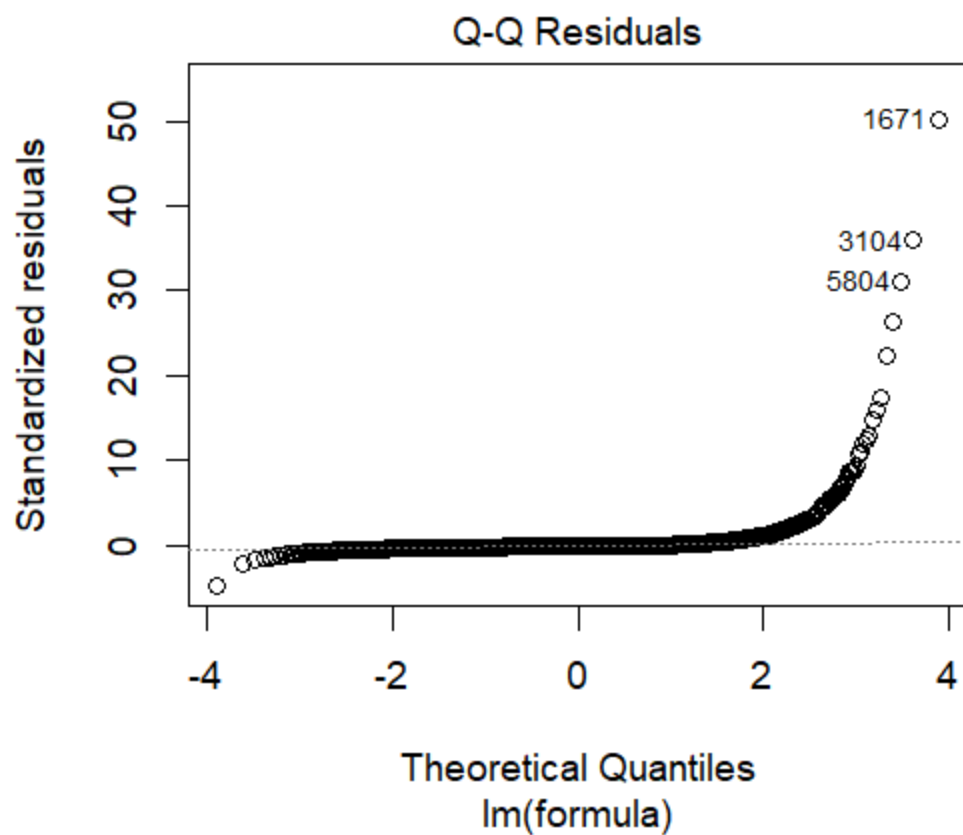
Le calcul du VIF montre aucune multicolinéarité préoccupante, bien que celles de Positive et Recommendations sont proches de 5.

### 3. ACF des résidus



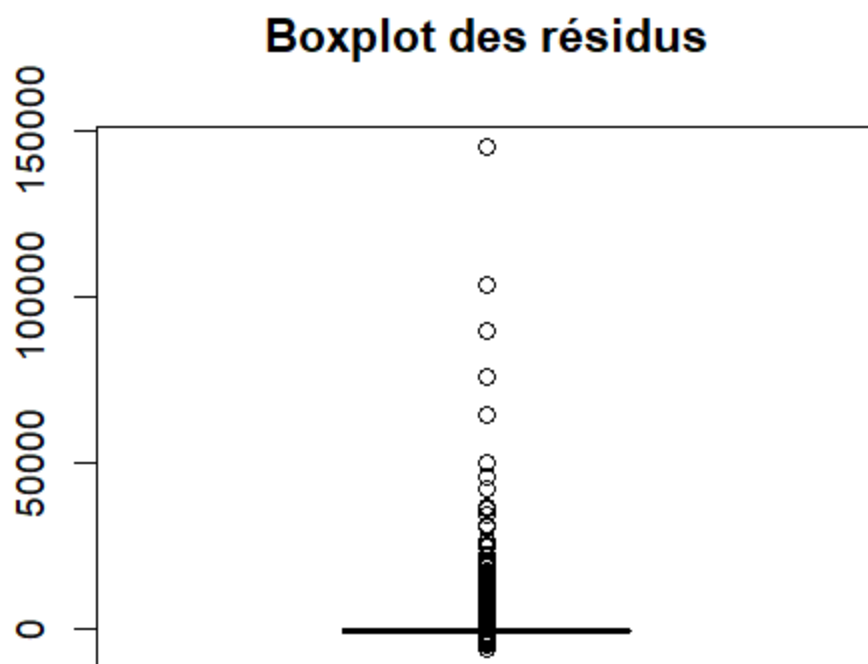
En accord avec le test de Durbin-Watson, les erreurs ne semblent pas auto-corrélées.

#### 4. QQ-plot des résidus

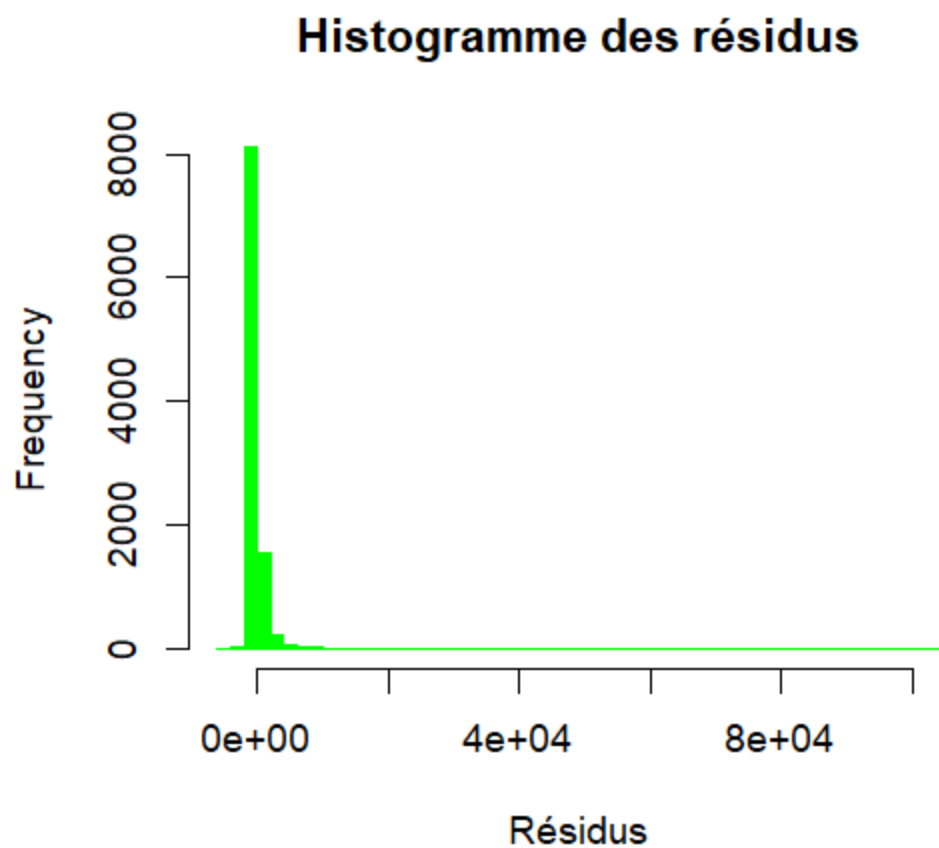


En cas de normalité des erreurs, on s'attend à ce que les points restent autour de la ligne, ce qui n'est pas le cas ici.



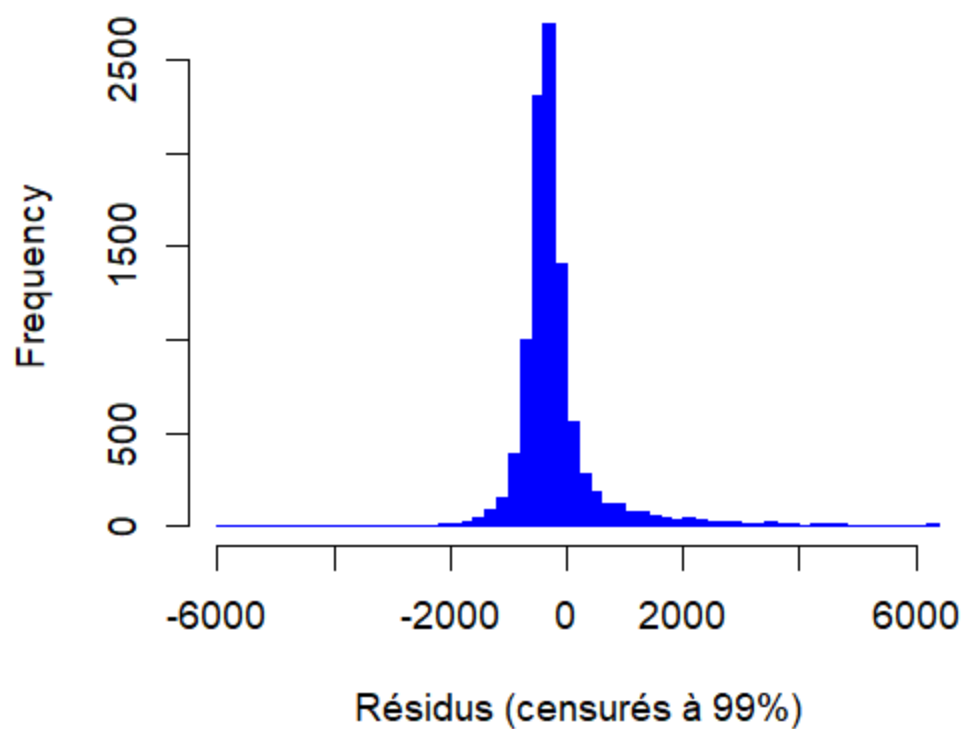


Boîte à moustache des résidus, la boîte compressée indique une majorité écrasante de résidus "faibles" avec des outliers particulièrement élevés.



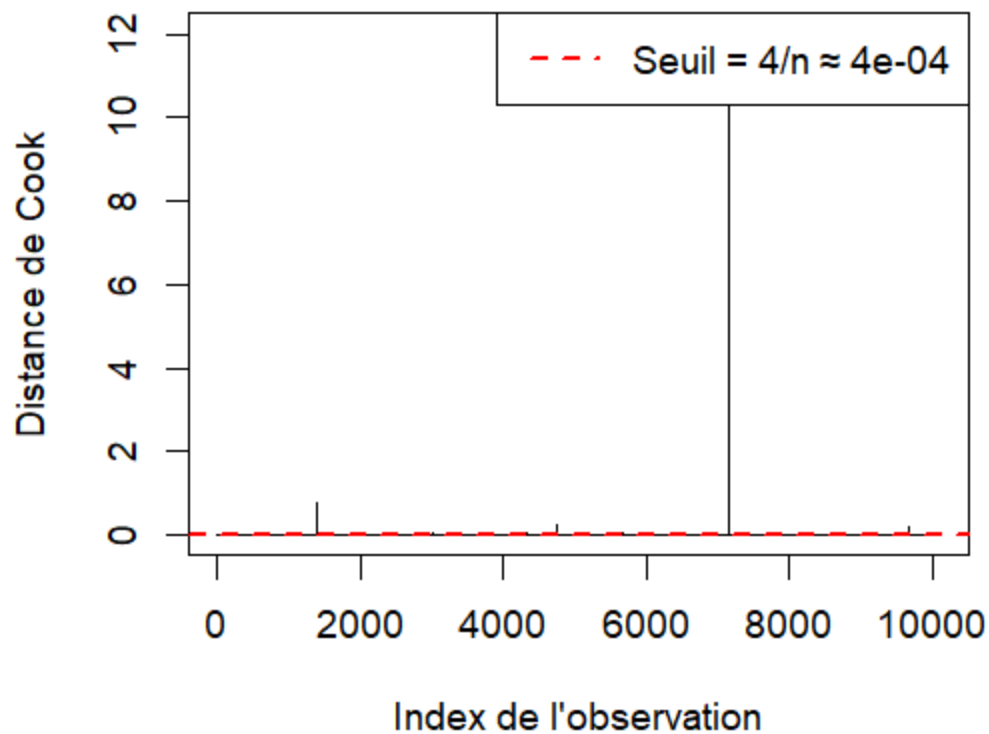
Autre visualisation des résidus, on voit bien la présence de résidus anormalement hauts.

## Histogramme des résidus (sans top 1%)



Même histogramme que précédemment, sans le top 1% des résidus les plus élevés.

## 6. Distance de Cook avec seuil $4/n$



Toutes les lignes dépassant la ligne rouge représentent des observations très influentes dans le modèle qui le perturberaient. On remarque une observation extrêmement influente sur le reste.

## Modèle 2 : transformation logarithmique

Pour réduire les effets des valeurs extrêmes, nous tentons de transformer certaines variables quantitatives avec un logarithme.

```
> model_log <- create_lm(gamesc_log, Y, X, categories)
> summary(model_log)
```

Call:  
lm(formula = formula, data = dataset)

Residuals:

Min	1Q	Median	3Q	Max
-2.19488	-0.28335	0.03172	0.32801	2.96211

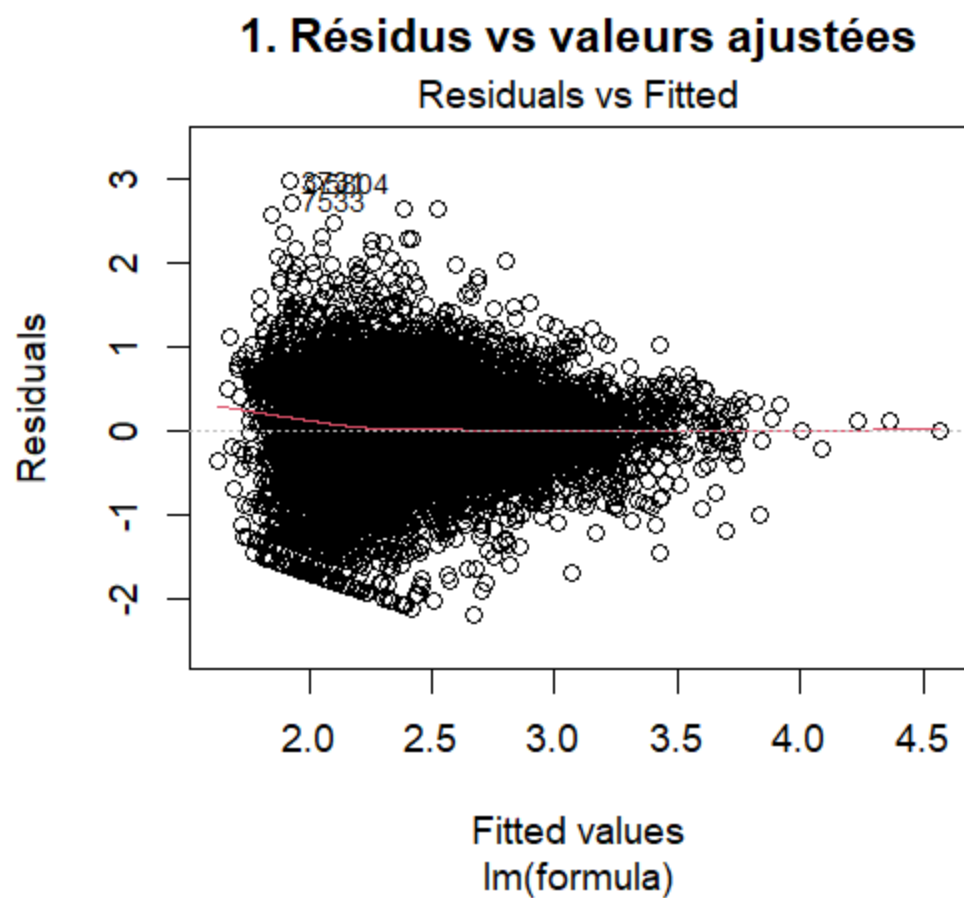
Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	1.464228	0.037749	38.789	< 2e-16	***
Peak.CCU	0.193272	0.008137	23.751	< 2e-16	***
Positive	0.058297	0.019536	2.984	0.002850	**
Negative	0.097102	0.015454	6.283	3.45e-10	***
Recommendations	0.013306	0.007476	1.780	0.075123	.
Price	0.157362	0.014342	10.972	< 2e-16	***
Required.age	-0.003604	0.001300	-2.773	0.005573	**
Estimated.owners100k-200k	0.104003	0.024301	4.280	1.89e-05	***
Estimated.owners1M-2M	0.193037	0.042247	4.569	4.95e-06	***
Estimated.owners10M-20M	0.331939	0.103763	3.199	0.001383	**
Estimated.owners100M-200M	1.048253	0.557391	1.881	0.060050	.
Estimated.owners20k-50k	0.067694	0.020660	3.277	0.001054	**
Estimated.owners200k-500k	0.134817	0.027489	4.904	9.52e-07	***
Estimated.owners2M-5M	0.221225	0.050582	4.374	1.23e-05	***
Estimated.owners20M-50M	0.391170	0.133806	2.923	0.003470	**
Estimated.owners50k-100k	0.082199	0.022267	3.691	0.000224	***
Estimated.owners500k-1M	0.193595	0.034748	5.571	2.59e-08	***
Estimated.owners5M-10M	0.297919	0.071976	4.139	3.51e-05	***
Estimated.owners50M-100M	0.706257	0.284366	2.484	0.013022	*

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

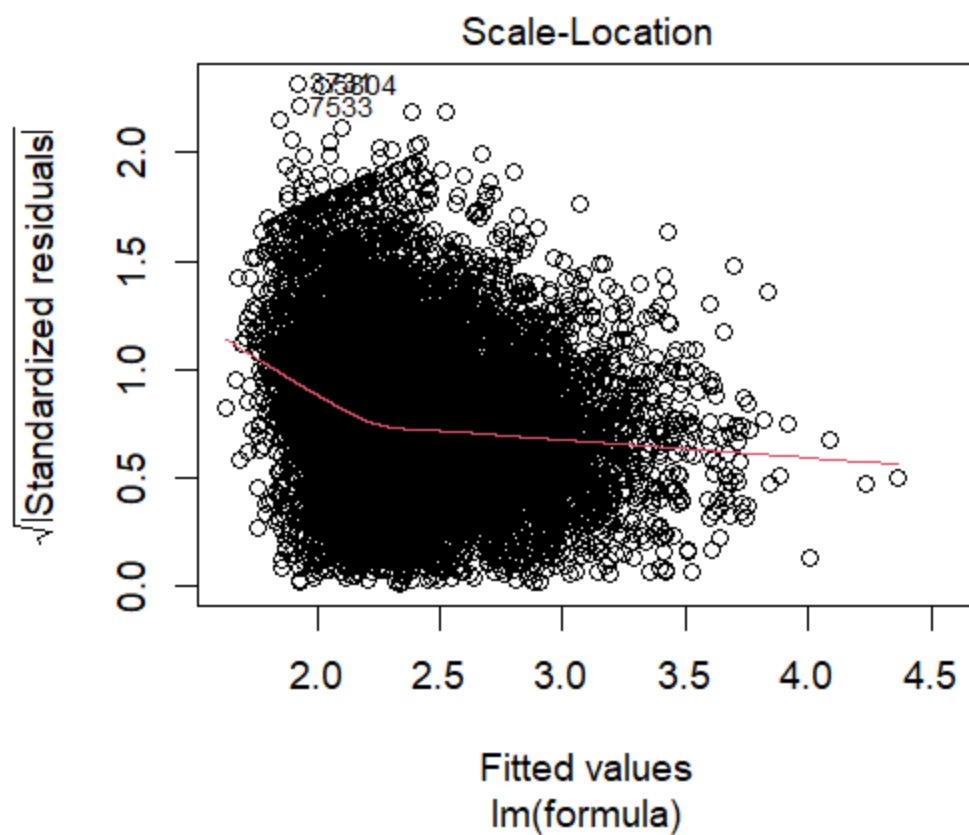
Residual standard error: 0.5523 on 10072 degrees of freedom  
Multiple R-squared: 0.2915, Adjusted R-squared: 0.2902  
F-statistic: 230.2 on 18 and 10072 DF, p-value: < 2.2e-16

Les coefficients de détermination sont bien meilleurs, presque à 30%, mais cela reste très bas.



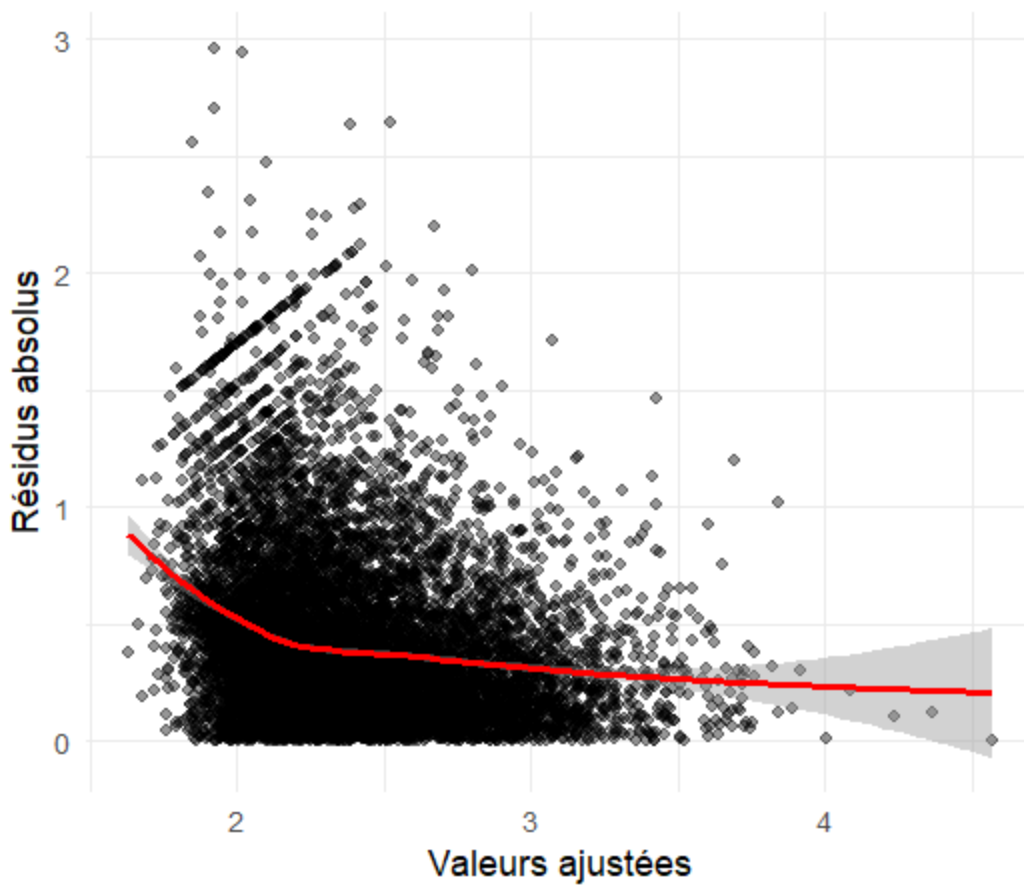
Bien meilleure répartition qu'avant, mais on a une forme d'entonnoir, compromettant ainsi l'hypothèse de linéarité.

## 2.1. Écarts à l'effet de levier



Meilleure répartition encore, mais forme d'entonnoir, donc problème d'homoscédasticité.

## 2.2. Hétéroscédasticité : résidus absolus vs ajustés



La ligne rouge montre que la variance des erreurs reste non constante selon les valeurs prédites.



```

> check_lm_hypotheses(model_log, gamesc_log)
Vérification des hypothèses pour le modèle : lm(formula = formula, data = dataset)
`geom_smooth()` using formula = 'y ~ x'
Test de Durbin-Watson (attendu ≈ 2) :

      Durbin-Watson test

data:  model
DW = 1.9403, p-value = 0.001339
alternative hypothesis: true autocorrelation is greater than 0

VIF (Variance Inflation Factor) :
      GVIF Df GVIF^(1/(2*Df))
Peak.CCU      2.046893   1      1.430697
Positive      7.726918   1      2.779733
Negative      4.298640   1      2.073316
Recommendations 3.215826   1      1.793272
Price          1.625266   1      1.274859
Required.age    1.113012   1      1.054994
Estimated.owners 4.863685  12      1.068129

Variables avec VIF > 5 :
NULL

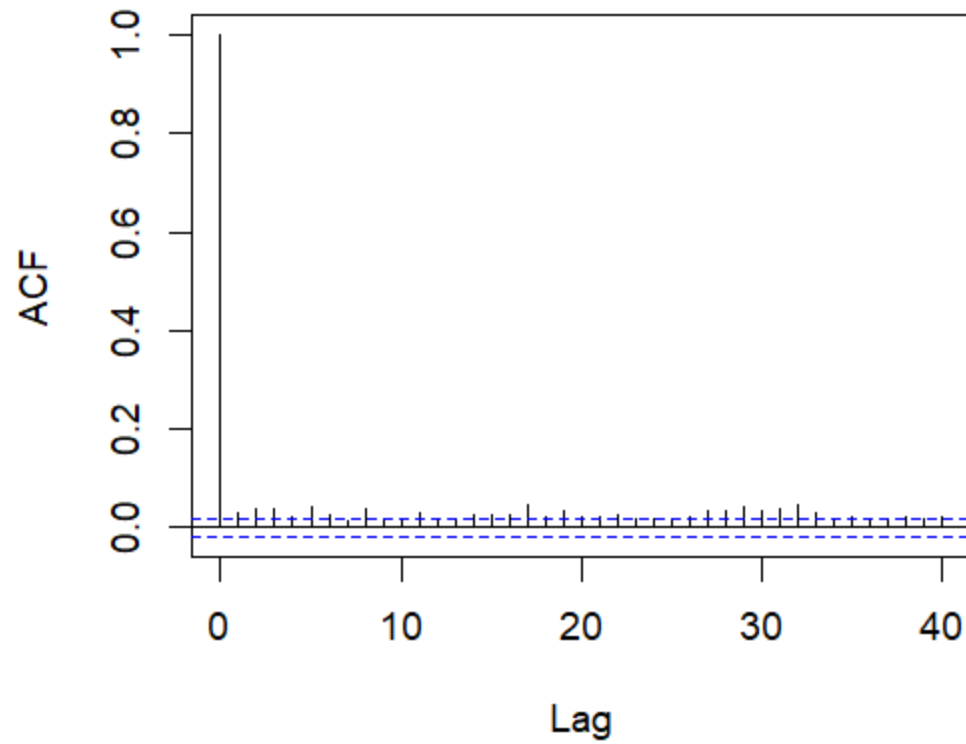
```

Le test de Durbin-Watson retourne une valeur proche de 2 mais une mauvaise p-value. Cela peut être dû au hasard ou à cause de sous groupes dans notre échantillon qu'il faudrait traiter indépendamment.

Les résultats sont donc incohérents, nous nous reposerons plutôt sur l'ACF.

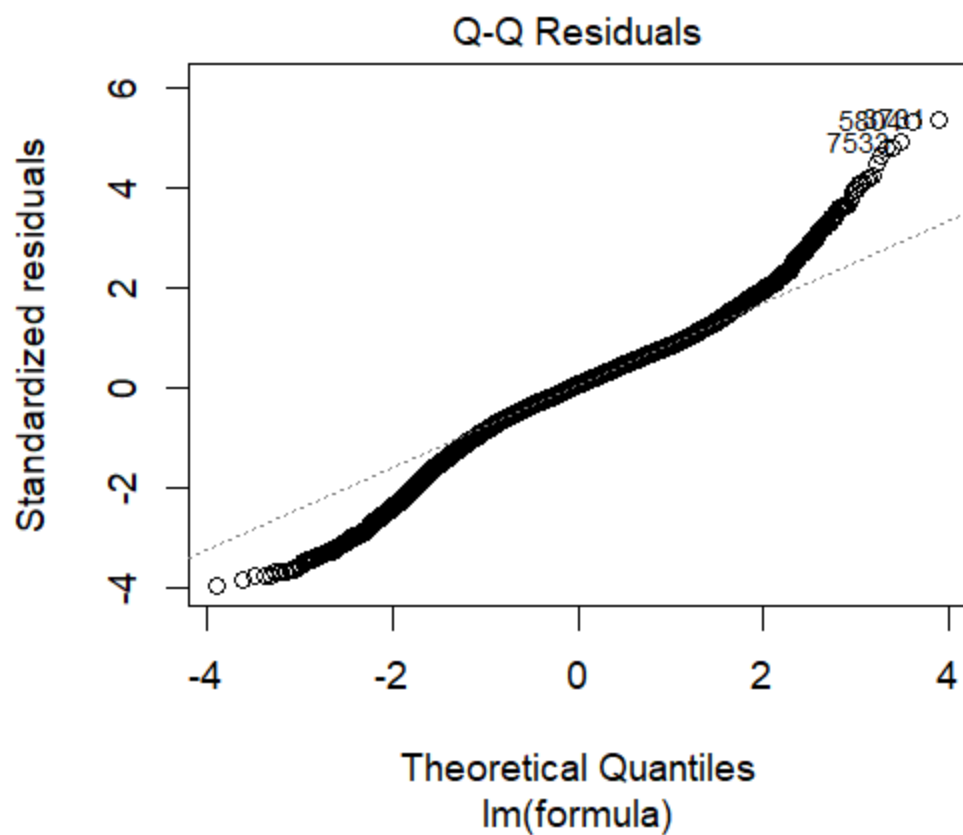
Le calcul du VIF comme avant ne montre aucune multicolinéarité inquiétante.

### 3. ACF des résidus



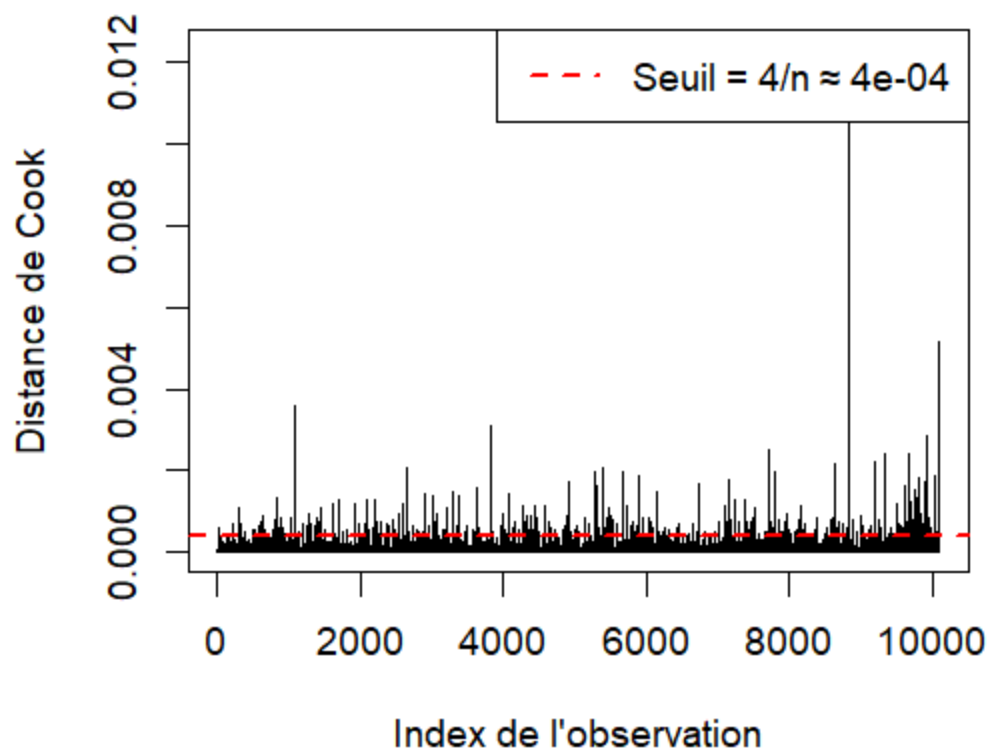
Presque la totalité des lignes dépassent les lignes bleues, montrant une grande autocorrélation des erreurs. Ce qui brise une hypothèse essentielle.

## 4. QQ-plot des résidus



Bien que l'allure de la courbe est légèrement meilleure, on reste loin de suivre la ligne droite. On rejette donc la normalité des erreurs.

## 6. Distance de Cook avec seuil $4/n$



La transformation logarithmique a fait apparaître un nombre inquiétant d'observations influentes, ce qui pourrait être encore une fois signe d'une sous structure que nos données ne contrôlent pas.

## Modèle 3 : log, sans outliers, sans les points de Cook, sans les résidus extrêmes

Le modèle précédent montrait des signes d'améliorations, mais brise d'autres hypothèses essentielles aux modèles linéaires. On tente ici de retirer les observations problématiques, les points trop influents, les outliers, et les résidus extrêmes.

```
> ## third model without outliers, high influence point, and extreme errors ----
> cleaning <- clean_model(model_log, gamesc_log)
> gamesc_log_clean <- cleaning$data
> model_log_clean <- create_lm(gamesc_log_clean, Y, X, categories)
> summary(model_log_clean)
```

Call:  
lm(formula = formula, data = dataset)

Residuals:

Min	1Q	Median	3Q	Max
-1.6918	-0.2732	0.0265	0.3045	1.5540

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	1.609080	0.035818	44.924	< 2e-16	***
Peak.CCU	0.219370	0.007950	27.593	< 2e-16	***
Positive	-0.016847	0.018899	-0.891	0.37273	
Negative	0.093637	0.014839	6.310	2.92e-10	***
Recommendations	0.021407	0.006998	3.059	0.00223	**
Price	0.175101	0.013435	13.033	< 2e-16	***
Required.age	-0.001938	0.004031	-0.481	0.63066	
Estimated.owners100k-200k	0.132481	0.022250	5.954	2.71e-09	***
Estimated.owners1M-2M	0.245090	0.040909	5.991	2.17e-09	***
Estimated.owners10M-20M	0.424485	0.173743	2.443	0.01458	*
Estimated.owners20k-50k	0.086280	0.018684	4.618	3.93e-06	***
Estimated.owners200k-500k	0.177488	0.025558	6.945	4.06e-12	***
Estimated.owners2M-5M	0.250928	0.050191	4.999	5.86e-07	***
Estimated.owners50k-100k	0.108160	0.020299	5.328	1.02e-07	***
Estimated.owners500k-1M	0.251289	0.032617	7.704	1.46e-14	***
Estimated.owners5M-10M	0.380371	0.085700	4.438	9.17e-06	***

----

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

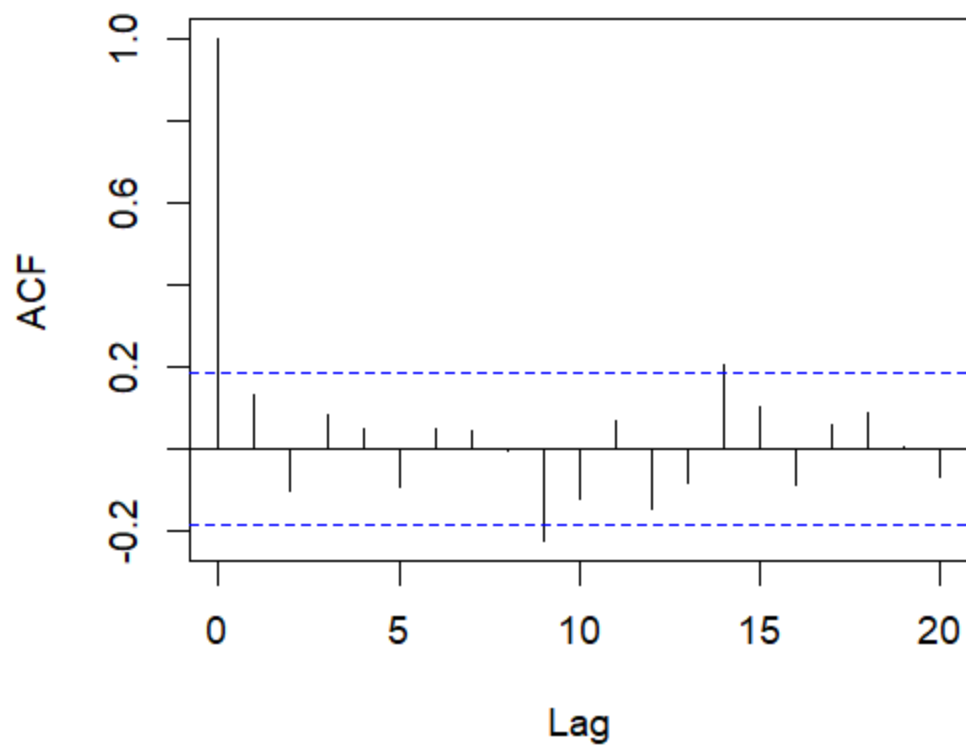
Residual standard error: 0.4755 on 8812 degrees of freedom  
Multiple R-squared: 0.302, Adjusted R-squared: 0.3008  
F-statistic: 254.2 on 15 and 8812 DF, p-value: < 2.2e-16

Il n'y a aux premiers abords aucun changement significatif par rapport au précédent modèle.

Vérifions maintenant si les tests d'hypothèses s'améliorent.

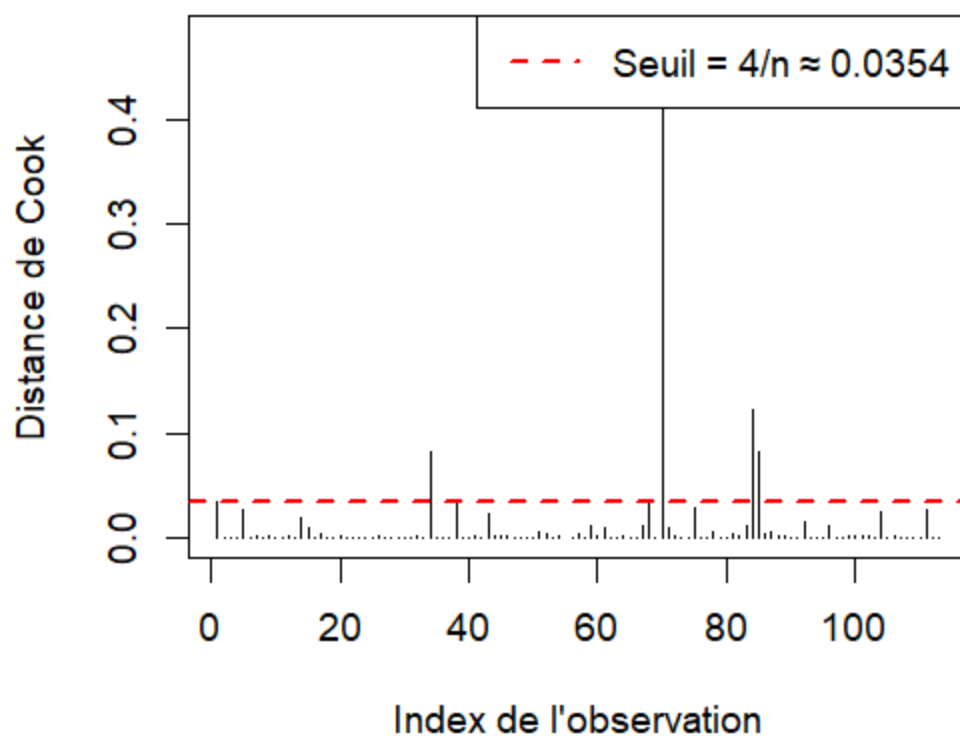


### 3. ACF des résidus



L'ACF des résidus montre beaucoup moins d'auto corrélation, une nette amélioration du modèle précédent.

## 6. Distance de Cook avec seuil $4/n$



Il reste quelques points influents ce qui n'est pas un problème en soi, mais malgré le nettoyage des points trop influents il y a encore une observation dont l'influence est anormalement haute, comme au premier modèle.



## Modèle sur les jeux Ubisoft

Comme nous supposons la présence de sous-populations, nous tentons de réduire l'analyse sur les jeux de l'éditeur Ubisoft.

```
> summary(model_ubisoft)

Call:
lm(formula = formula, data = dataset)

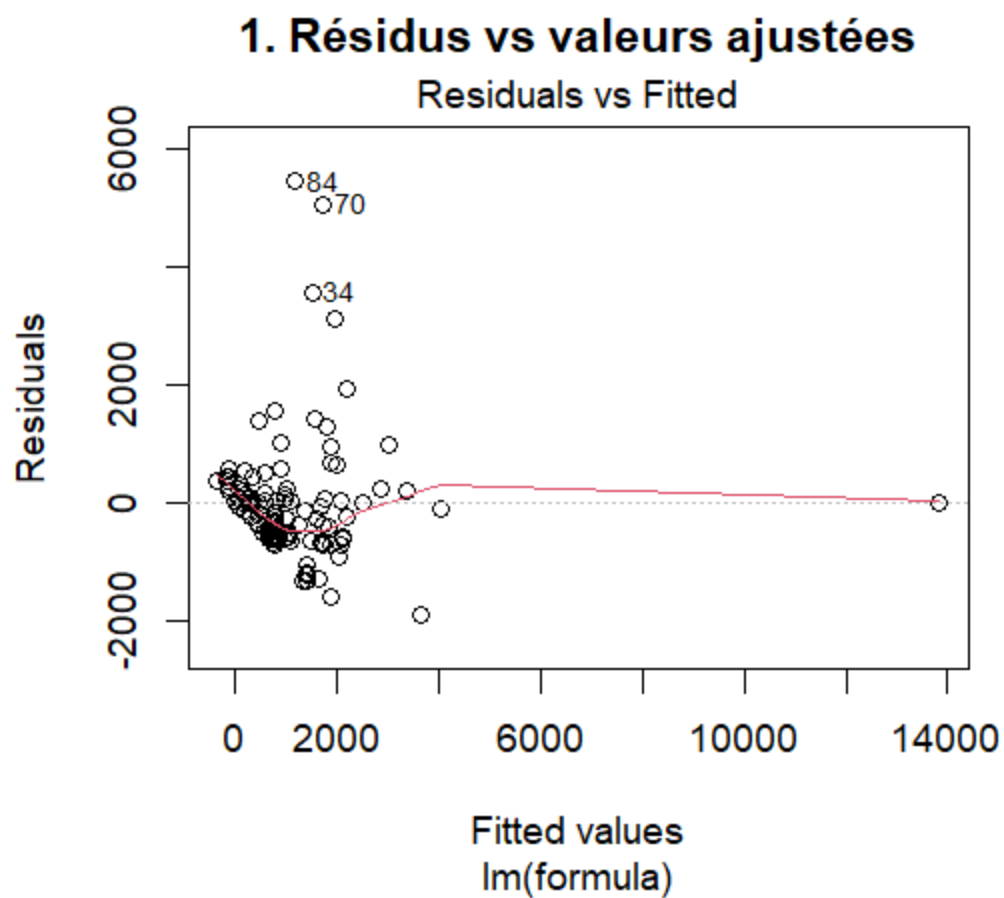
Residuals:
    Min       1Q   Median       3Q      Max
-1905.5  -548.1  -155.5   197.4   5454.5

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.164e+02  8.776e+02  0.133   0.8948
Peak.CCU     4.039e-02  6.152e-02  0.657   0.5130
Recommendations 1.172e-02  2.519e-03  4.651 1.05e-05 ***
Price        1.910e+01  1.027e+01  1.859   0.0660 .
Required.age -3.511e+01  1.513e+01 -2.321   0.0224 *
Estimated.owners100k-200k -1.687e+02  8.651e+02 -0.195   0.8458
Estimated.owners1M-2M    6.286e+02  9.071e+02  0.693   0.4900
Estimated.owners20k-50k  7.884e+02  9.607e+02  0.821   0.4139
Estimated.owners200k-500k 1.387e+02  8.492e+02  0.163   0.8706
Estimated.owners2M-5M    9.633e+02  9.105e+02  1.058   0.2927
Estimated.owners20M-50M  1.363e+03  1.764e+03  0.773   0.4415
Estimated.owners50k-100k -1.944e+02  8.833e+02 -0.220   0.8263
Estimated.owners500k-1M  1.021e+03  8.754e+02  1.167   0.2463
rating.L      9.417e+02  9.300e+02  1.013   0.3138
rating.Q     -9.960e+02  7.746e+02 -1.286   0.2016
rating.C      6.321e+02  4.927e+02  1.283   0.2026
rating^4      9.844e+01  2.749e+02  0.358   0.7210
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1140 on 96 degrees of freedom
Multiple R-squared:  0.6575,    Adjusted R-squared:  0.6005
F-statistic: 11.52 on 16 and 96 DF,  p-value: 5.341e-16
```

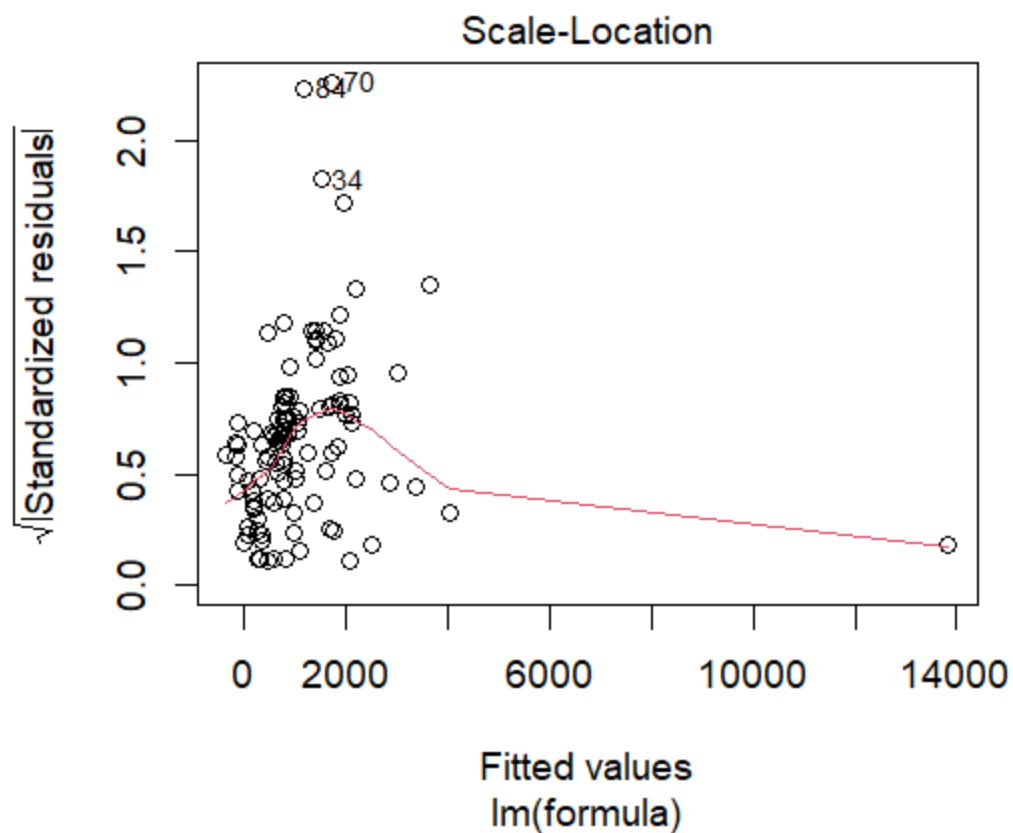
Nous remarquons enfin un  $R^2$  et un  $R^2$  ajusté passable, au-dessus de 60%, les résidus semblent nettement plus bas aussi, comparé au modèle naïf. Les p-value restent basses, affirmant la significativité du modèle.

Vérifions maintenant comment se portent les hypothèses.



Le nuage de points reste irrégulier, au point de provoquer une forte courbure de la ligne rouge. On rejette encore une fois l'hypothèse de linéarité.

## 2.1. Écarts à l'effet de levier



Le nuage de point irrégulier remet encore en cause l'homoscédasticité des erreurs. La restriction de l'analyse à Ubisoft maintient cette concentration écrasante de valeurs faibles.

```

> check_lm_hypotheses(model_ubisoft, ubisoft)
Vérification des hypothèses pour le modèle : lm(formula = formula, data = dataset)

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

Test de Durbin-Watson (attendu ≈ 2) :

      Durbin-Watson test

data:  model
DW = 1.7014, p-value = 0.04713
alternative hypothesis: true autocorrelation is greater than 0

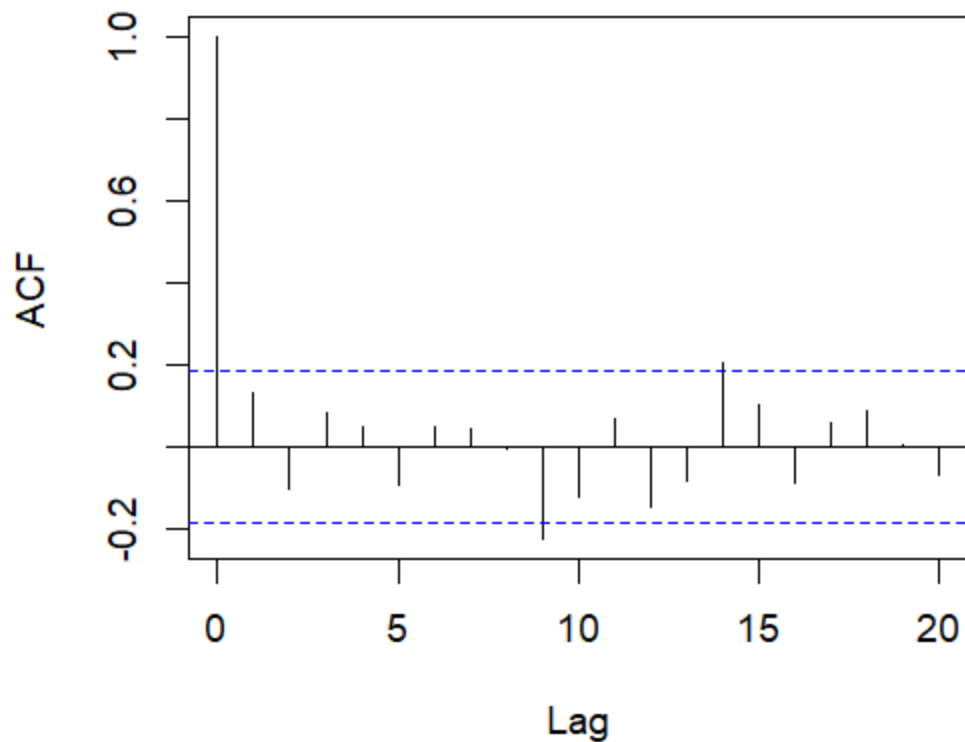
VIF (Variance Inflation Factor) :
      GVIF Df GVIF^(1/(2*Df))
Peak.CCU      7.036400  1      2.652621
Recommendations 4.066558  1      2.016571
Price          1.566473  1      1.251588
Required.age    1.411481  1      1.188058
Estimated.owners 9.164647  8      1.148503
rating          1.464069  4      1.048806

Variables avec VIF > 5 :
NULL

```

Le test de Durbin-Watson renvoie 1,7 et une mauvaise p-value trop faible. Ce qui validerait l'autocorrélation de premier ordre.

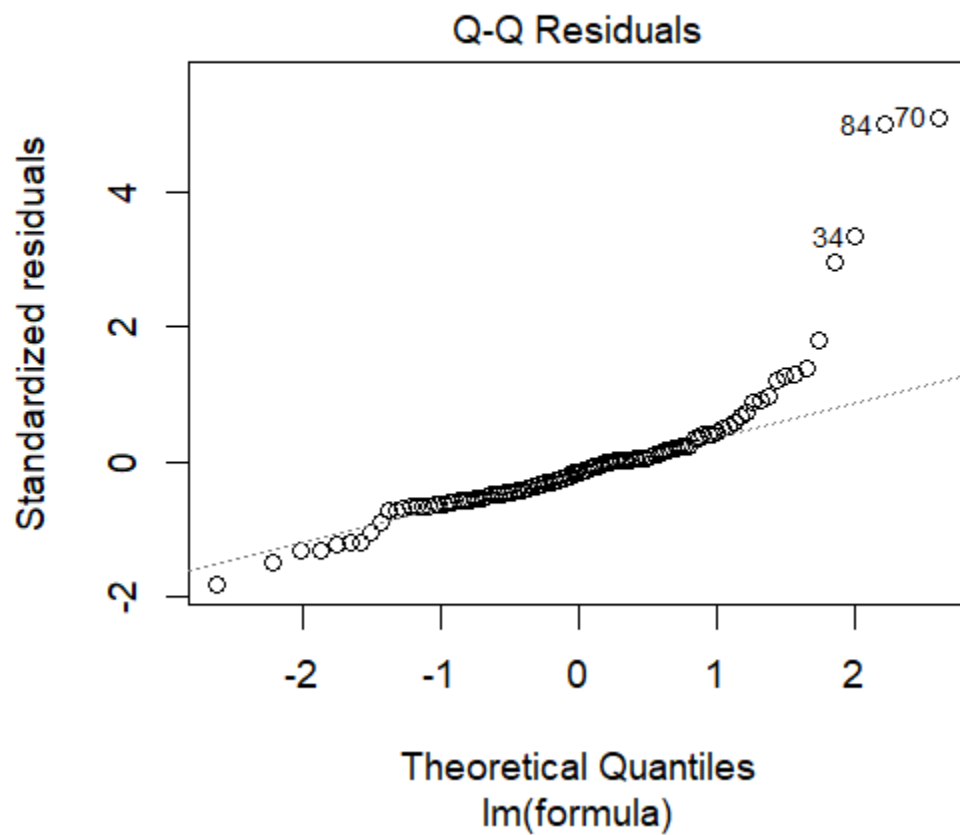
### 3. ACF des résidus



Malgré le test de Durbin-Watson, l'ACF ne détecte pas une autocorrélation des erreurs importante. Comme nous n'avons aucune temporalité dans les variables analysées, nous décidons de nous fier à l'ACF.

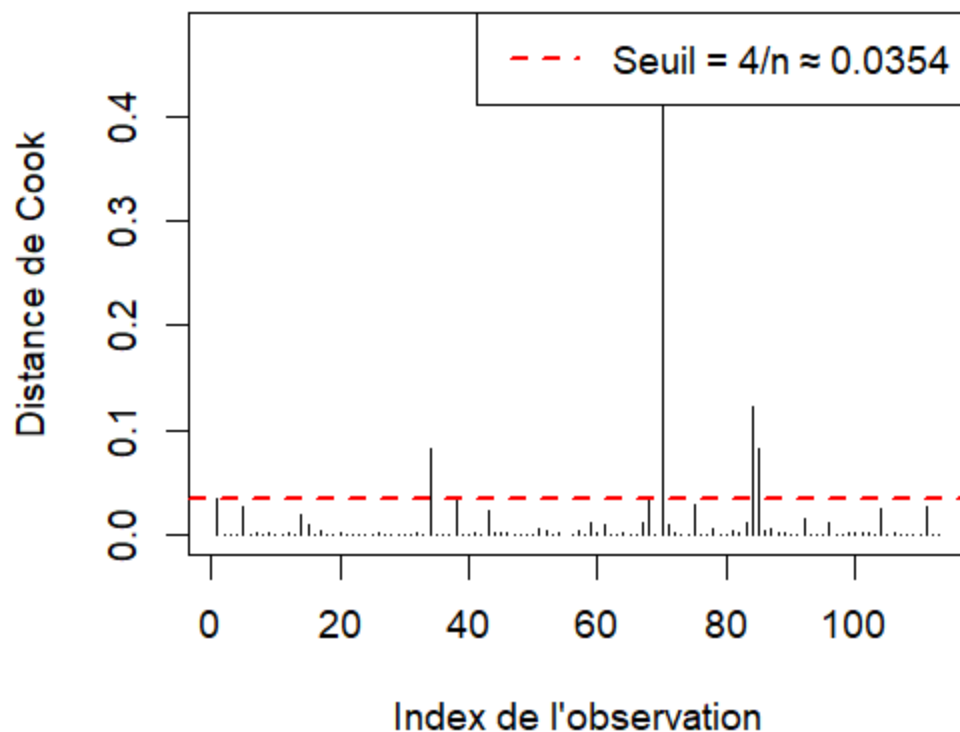
On aurait donc une légère autocorrélation des erreurs, indiquée par les 2 barres qui dépassent la zone de confiance.

#### 4. QQ-plot des résidus



Le nuage de points ne suit pas la ligne, rejetant donc la normalité des erreurs.

## 6. Distance de Cook avec seuil 4/n



Quelques points très influents, dont un qui est particulièrement perturbant. Peut-être un jeu d'Ubisoft qui a été exceptionnellement populaire.

Index	Average.playtime.forever	Estimated.owners	Peak.CCU	rating
17	13837	20M-50M	42263	Very Positive
70	6808	20k-50k	21	Mostly Positive
84	6630	200k-500k	158	Very Positive
85	5115	2M-5M	560	Mixed
34	5091	500k-1M	329	Very Positive

En regardant de plus près les données, on remarque que les deux jeux au plus haut Average.playtime.forever semblent très inhabituels vu les disparités entre Peak.CCU et Average.playtime.forever.

## Algorithme de sélections de modèle linéaire sur les jeux ubisoft

Le premier modèle naïf des jeux Ubisoft s'étant montré assez prometteur en termes de performance explicative, nous décidons d'essayer de raffiner le modèle via des algorithmes de sélection (malgré certaines hypothèses non validées).

```
> # backward direction returns trivial model, so we are not comparing them
> compare_models(models_to_compare, names_to_use)
      Model      AIC      BIC Adj_R2      RSE Shapiro.p      BP.p Mean_VIF
BP      AIC 1918.98 1935.34 0.598 1143.27      0 0.5706      4.18
BP1     BIC 1920.59 1934.23 0.589 1156.34      0 0.3786      1.04
BP2     AIC forward 1918.98 1935.34 0.598 1143.27      0 0.5706      4.18
BP3     BIC forward 1920.59 1934.23 0.589 1156.34      0 0.3786      1.04
BP4 Fischer forward 1918.98 1935.34 0.598 1143.27      0 0.5706      4.18
BP5     AIC both 1918.98 1935.34 0.598 1143.27      0 0.5706      4.18
BP6     BIC both 1920.59 1934.23 0.589 1156.34      0 0.3786      1.04
BP7     Fischer both 1918.98 1935.34 0.598 1143.27      0 0.5706      4.18
>
> ## show variables used in each model
> for (i in seq_along(models_to_compare)) {
+   cat("\n---", names_to_use[i], "---\n")
+   print(attr(terms(models_to_compare[[i]]), "term.labels"))
+ }

--- AIC ---
[1] "Price"          "Recommendations" "Required.age"     "Negative"

--- BIC ---
[1] "Price"          "Required.age"    "Negative"

--- AIC forward ---
[1] "Negative"       "Price"          "Required.age"     "Recommendations"

--- BIC forward ---
[1] "Negative"       "Price"          "Required.age"

--- Fischer forward ---
[1] "Negative"       "Price"          "Required.age"     "Recommendations"

--- AIC both ---
[1] "Negative"       "Price"          "Required.age"     "Recommendations"

--- BIC both ---
[1] "Negative"       "Price"          "Required.age"

--- Fischer both ---
[1] "Negative"       "Price"          "Required.age"     "Recommendations"
>
```

Tout les algorithmes renvoient des modèles similaires, les 4 mêmes variables sont gardées à chaque fois: Negative, Price, Required.age et Recommendations.

Ce qui est relativement suprenant vu les suppositions que l'on pouvait se faire à l'analyse des corrélations, nous pensions que Peak.CCU et Positive feraient partie des variables importantes.



La méthode BIC privilégie un modèle plus simple en retirant Recommendations pour un  $R^2$  ajustée légèrement plus bas.

La méthode step by step backward renvoyait le modèle trivial, donc ils ne sont pas présent dans la comparaison.

## Classification/Modèle logistique et polytomique

Pour cette section la variable que nous souhaitons expliquer est Estimated.owners, au lieu de Average.playtime.forever.

Au vu des résultats des modèles linéaires précédents nous appliquons sur les valeurs une standardisation et un logarithme.

```
> # testing coefficient significance
> z <- summary(model_logit)$coefficients / summary(model_logit)$standard.errors
> p_values <- 2 * (1 - pnorm(abs(z)))
> cat("\n--- P-values des coefficients ---\n")

--- P-values des coefficients ---
> print(round(p_values, 4))
```

	(Intercept)	Average.playtime.forever	Peak.CCU	Positive	Negative	Recommendations	Price	Required.age
20k-50k	0e+00	0.0004	0.3022	0e+00	0	0.0000	0e+00	0.1610
50k-100k	0e+00	0.0000	0.0000	0e+00	0	0.0000	0e+00	0.0069
100k-200k	0e+00	0.0000	0.0000	0e+00	0	0.0000	0e+00	0.0160
200k-500k	0e+00	0.0000	0.0000	0e+00	0	0.0000	0e+00	0.0009
500k-1M	0e+00	0.0000	0.0000	0e+00	0	0.0000	0e+00	0.0000
1M-2M	0e+00	0.0000	0.0096	0e+00	0	0.0000	0e+00	0.0000
2M-5M	0e+00	0.0000	0.0169	0e+00	0	0.0000	0e+00	0.0000
5M-10M	0e+00	0.0000	0.2443	0e+00	0	0.0000	0e+00	0.0000
10M-20M	0e+00	0.0004	0.5275	0e+00	0	0.0000	0e+00	0.0277
20M-50M	0e+00	0.8972	0.0000	0e+00	0	0.0000	0e+00	0.0002
50M-100M	0e+00	0.8231	0.0001	0e+00	0	0.0000	0e+00	0.6377
100M-200M	9e-04	0.1622	0.4027	1e-04	0	0.0226	5e-04	0.0935

Quelques mauvaises p-values mais pas de pattern évident, à part que Average.playtime.forever perd en significativité pour les classes hautes.

```
> cat("\n--- VIF (multicolinéarité) ---\n")


--- VIF (multicolinéarité) ---
> print(vif(mod_lineaire_temp))
```

Peak.CCU	Positive	Negative	Recommendations	Price	Required.age
1.984258	6.000527	3.640257	3.005692	1.527625	1.088107

Le VIF ne montre une multicolinéarité importante pour Positive, non étonnant car on le sait fortement corrélé à Negative et Recommendations.

```
--- Taux de bonnes prédictions ---
> print(round(accuracy, 4))
[1] 0.44
> cat("\n--- Taux de bonnes prédictions (modèle trivial) ---\n")

--- Taux de bonnes prédictions (modèle trivial) ---
> print(round(trivial_accuracy, 4))
[1] 0.1786
> |
```



Le modèle logistique trivial a une précision de 17% tandis que le modèle testé est à 44%. Il apporte donc une certaine valeur malgré les hypothèses de linéarité manquantes et la sous représentation de certaines classes.