PROJET RÉALISÉ PAR L'ÉQUIPE

# GROUPE 17 VIDÉO GAMES ANALYTICS

## RAPPORT DE GROUPE EN SCIENCES DES DONNÉES 2 + BASES DE DONNÉES

Anthony COMBES–AGUÉRA (22113542), Mohamed Chaouki REKHIS (22212667),
Raphaël BAYET (22206475), Yanick TINAUT (22212676)


Département MIASHS, UFR 6 Informatique, Mathématique et Statistique
Université Paul Valéry, Montpellier 3

Avril 2024

Soumis comme contribution partielle
pour le cours Science des données 2 et Bases de données

# Déclaration de non plagiat

Nous déclarons que ce rapport est le fruit de notre seul travail, à part lorsque cela est indiqué explicitement.

Nous acceptons que la personne évaluant ce rapport puisse, pour les besoins de cette évaluation:

- la reproduire et en fournir une copie à un autre membre de l'université; et/ou,
- en communiquer une copie à un service en ligne de détection de plagiat (qui pourra en retenir une copie pour les besoins d'évaluation future).

Nous certifions que nous avons lu et compris les règles ci-dessus.

En signant cette déclaration, nous acceptons ce qui précède.

Signature: _____ Date: _____

Signature: _____ Date: _____

Signature: _____ Date: _____

Signature: _____ Date: _____

# Remerciements

Nos plus sincères remerciements vont à notre encadrant pédagogique pour les conseils avisés sur notre travail.

Nous remercions aussi Ayoub AKKOUH pour son aide psycologique durant cette période compliquée pour nous.

30/04/2024.

# Résumé

Dans notre rapport, nous avons examiné l'influence du prix initial et des critiques sur le nombre de joueurs actifs pour des jeux sur Steam. Notre analyse portant sur 950 jeux depuis leur lancement jusqu'en juillet 2022 a révélé que ceux notés au-dessus de 75% par la presse voient leur base de joueurs augmenter en moyenne de 20%, indépendamment du prix. Pour les jeux à moins de 20 euros, un score élevé peut quadrupler le nombre de joueurs actifs durant le premier mois après la sortie.

Ces observations démontrent l'importance des critiques, surtout pour les jeux à bas prix. Les stratégies de tarification et de marketing doivent donc être ajustées en prévision de l'accueil des évaluateurs. Pour approfondir, des analyses futures pourraient explorer l'effet des genres de jeux ou des mises à jour post-lancement sur l'engagement des joueurs.

Nous préconisons deux perspectives : à court terme, améliorer l'exactitude des prévisions du nombre de joueurs en intégrant plus de variables; à long terme, étudier l'impact des tendances de l'industrie du jeu, telles que les jeux en tant que service, les abonnements, les microtransactions au sein d'un jeu initialement gratuit.

Nos difficultés résidaient principalement dans le choix et la manipulation des bases de données. Les changements fréquents de dataset et les ajustements de notre question de recherche ont été motivés par la quête de données adaptées à nos besoins. Le partage des données post-traitement a également présenté des défis, résolus en optant pour GitHub, malgré son interface peu intuitive au départ. Le nettoyage des données n'a pas posé de problème significatif, mais la standardisation des noms de jeux et la conversion des dates en valeurs numériques ont demandé un effort considérable. Enfin, l'apprentissage et l'application des concepts théoriques, comme la rédaction en R Markdown, ont exigé une grande dynamique d'apprentissage de notre part.

Ce projet a été une opportunité d'approfondir nos compétences pratiques en science des données et nous a permis de réaliser un rendu qui nous tient à cur.

# Table des matières

# Chapitre 1

# Introduction

## 1.1 Présentation

Dans une ère qui se dirige de plus en plus vers le numérique, le secteur des jeux vidéo est plus que jamais au cur des transformations économiques et culturelles. Alors que le monde du réel et celui du virtuel tendent à se confondre, l'univers du jeu vidéo, permet d'accroître de manière exponentielle, l'expérience récréative que peuvent ressentir les joueurs.

De ses modestes débuts, dans les années 1950, à son statut actuel de géant du divertissement, l'industrie du jeu vidéo a su faire preuve d'une innovation constante dans la manière dont l'humanité interagit avec le numérique, afin de devenir l'un des secteurs le plus lucratif de l'économie mondiale.

Enfin, pour les développeurs et les distributeurs, engendrer des revenus substantiels et maintenir une communauté de joueurs fidèles sont devenues des objectifs centraux lorsqu'il s'agit de produire un jeu. Face à cette réalité, notre projet ce concentre sur l'impact que peut avoir le prix et les appréciations de la critique d'un jeu vidéo sur la popularité de celui-ci.

Notre question de recherche est la suivante :

**Comment le nombre de joueurs actifs pour un jeu évolue-t-il par rapport à son prix initial et à la note que la presse lui attribue, au sein de la communauté de Steam ?**

Ainsi, il s'agirait de savoir si un jeu vidéo ayant un prix initial faible et une note élevée, permettrait de mieux maintenir une base de joueur actifs dans le temps, que les autres jeux.

L'importance de cette question réside dans son implication pour l'industrie du jeu vidéo, un secteur qui continue de croître et d'évoluer à une vitesse vertigineuse. Comprendre ce que les critiques des différents médias pourraient apporter sur les ventes et le choix des joueurs. Ceci permettrait d'offrir des insights précieux pour les développeurs et les marketeurs dans leur stratégie de création, de vente et de promotion de leurs jeux. En outre, cette recherche peut éclairer les consommateurs sur les dynamiques qui influencent la popularité et le succès commercial des jeux vidéo.

## 1.2 Responsabilités et composition de l'équipe

Nous nous sommes réparti les rôles pour couvrir tous les aspects nécessaires à l'accomplissement de ce projet, de la collecte des données à la rédaction du rapport.

- **Yanick Tinaut** a découvert le premier jeu de données intitulé 'Steam Games' sur le site de Hugging Face. Il a également rédigé la première version (la V1) de la partie consacrée au cours de base de données sur le site Overleaf. Il a rédigé l'ensemble des requêtes SQL. Avec **Anthony Combes-Aguera**, ils ont pris en charge la rédaction des chapitres 2 et 3, qui traitent des jeux de données, ainsi que des sections sur l'interprétation des résultats, la conclusion et les perspectives que pourrait offrir cette étude.
- **Anthony Combes-Aguera** s'est quant à lui, occupé du prétraitement et du nettoyage des jeux de données à l'aide de la plateforme R Studio. Il a également importé les jeux de données sur la plateforme phpMyAdmin et implémenté les requêtes SQL directement dans le fichier Rmarkdown. Avec **Yanick Tinaut**, ils ont écrit les commentaires et les interprétations de ces résultats ainsi que la correction totale de l'ensemble des graphiques. Anthony est également le créateur de notre diaporama et de son contenu.
- **Raphaël Bayet** a découvert le second jeu de données 'Steam Chart' sur le site Kaggle. Avec **Mohamed Rekhis Chaouki**, ils ont réalisé l'analyse statistique descriptive. Ils ont créé les graphiques et rédigé les interprétations correspondantes, puis les ont inclus dans le fichier Rmarkdown final.
- **Mohamed Rekhis Chaouki** s'est vu chargé de produire le test ANOVA.

Malgré le fait que nous avions réparti les différentes tâches entre les membres, nous nous sommes entraidés sur les différents aspects du projet, ce qui a non seulement rendu notre travail plus agréable, mais également rendu celui-ci, beaucoup plus solide et cohérent.

# Chapitre 2

# Base de données

## 2.1 Provenance des données

Pour notre étude sur l'impact des évaluations et du prix initial des jeux vidéo sur la longévité du nombre de joueurs actifs sur ceux-ci, nous avons utilisé deux principaux jeux de données, disponibles en ligne sur le site **Kaggle**[1] et **Hugging Face**[2].

1. **Steam Games Dataset**[3]: Fournit une vue d'ensemble complète des jeux disponibles sur la plateforme Steam, en se concentrant sur des aspects tels que le prix de lancement, les genres, les notes des utilisateurs et de la presse, la moyenne de temps de jeu, et d'autres métadonnées essentielles. Ces informations permettent d'explorer les facteurs contribuant au succès initial et à la perception générale d'un jeu dans l'écosystème de Steam. En particulier, le prix de lancement peut être un indicateur crucial de la stratégie de positionnement d'un jeu sur le marché, tandis que les notes de Metacritic reflètent la réception communautaire et la satisfaction à l'égard de l'expérience de jeu.
2. **Player Counts on Steam**[4]: Offre des données sur le nombre moyen de joueurs actifs par mois pour différents jeux sur Steam, permettant d'évaluer la fidélité ou la longévité du nombre de joueurs, dans le temps, pour un jeu donné. Ces valeurs sont des indicateurs clés de la réussite à long terme d'un jeu. En croisant ces données avec les notes et le prix des jeux vidéo, il est peut-être possible d'analyser comment ces deux variables affectent la capacité d'un jeu à maintenir une base de joueurs active dans le temps.

Ces bases de données ont été choisies pour leur complémentarité, permettant une analyse approfondie de l'impact des évaluations et de son prix d'achat sur la popularité à long terme des jeux vidéo. Nous avons filtré les données pour nous concentrer sur la période de 2012 à 2020, en ne sélectionnant que les jeux disposant des données de nombre de joueurs actifs uniquement après leur date de sortie. La population ciblée inclut donc les jeux vidéo lancés durant cette période, avec une attention particulière portée aux titres ayant des données qui sont cohérentes aux deux dataset. Pour réaliser cela, nous avons prétraité les données des deux bases de données initiales avec le logiciel R Studio.

[1] Kaggle, disponible sur https://www.kaggle.com/datasets.

[2] Hugging Face, disponible sur https://huggingface.co.

[3] Steam Games Dataset sur Hugging Face, disponible sur https://huggingface.co/datasets/FronkonGames/steam-games-dataset.

[4] Player Counts on Steam sur Kaggle, disponible sur https://www.kaggle.com/datasets/josephvm/player-counts-on-steam.

## 2.2 Descriptif des tables

1. **Steam Games Dataset**: Ce jeu de données se présente sous la forme d'un fichier CSV de 244 Mo, comprenant 85 104 entrées, chacune représentant un jeu unique avec des informations détaillées réparties sur 39 colonnes. En combinant ces données avec des informations sur le nombre de joueurs actifs, il pourrait être possible d'analyser des tendances comme l'impact du prix et de la satisfaction des utilisateurs sur la longévité et la popularité d'un jeu au sein de la communauté Steam.
2. **Player Counts on Steam**: Les données sont présentées dans un fichier CSV de 3.49 Mo comportant 999 lignes qui représentent des jeux uniques et 7 colonnes.

## 2.3 Description du nettoyage des données

Le nettoyage des données a été fait sur RStudio[5].

### *2.3.1 Bibliothèques utilisées*

- `dplyr` : utilisée pour la manipulation des data frames.
- `stringr` : permet le traitement efficace des chaînes de caractères.
- `lubridate` : simplifie la gestion des dates.

### *2.3.2 Chargement des données*

Les données sont chargées depuis des fichiers CSV locaux. Les principaux ensembles de données sont :

1. `games.csv` : contient les détails des jeux.
2. `steam_charts.csv` : inclut les informations sur les comptages de joueurs.

### *2.3.3 Renommage et préparation des colonnes*

Les colonnes de `steam_charts` ont été renommées pour clarifier leur usage (`App.ID` en `AppID` et `Game` en `Name`). Les transformations incluent la conversion des pourcentages en valeurs numériques et le remplacement des tirets par des zéros pour corriger les erreurs de type.

### *2.3.4 Filtrage des jeux*

Nous avons exclu les jeux en chinois simplifié pour nous concentrer sur ceux ayant des titres en alphabet latin, ce qui était essentiel pour garantir l'intégrité des données dans `steam_charts`. La date de sortie a été uniformisée pour faciliter les comparaisons.

### *2.3.5 Mise en commun des jeux*

Un nettoyage supplémentaire a permis d'aligner les jeux dans nos fichiers CSV en utilisant l'identifiant `AppID`.

### *2.3.6 Sélection des colonnes*

Nous avons réalisé une sélection rigoureuse des colonnes, conservant uniquement celles pertinentes pour l'analyse ultérieure. En tout, 30 variables ont été supprimées du 'Steam Games Dataset'. Pour 'Player Counts on Steam', seule la variable `Game` a été supprimée, car `AppID` sert de clé primaire plus efficace.

### *2.3.7 Organisation finale et sauvegarde*

Les données finales ont été organisées par `AppID` et sauvegardées au format CSV pour une utilisation future dans l'analyse.

[5]Le code est disponible en annexe.

### *2.3.8 Tableaux synthèse*

Table 2.1: Player Counts on Steam (950 x 9)

| Nom.colonne | Type | Signification | Caractéristique |
|---|---|---|---|
| AppID | Integer | Identifiant du jeu sur Steam | Clé primaire, unique |
| Name | String | Nom du jeu | Pas de valeurs manquantes |
| ReleaseDate | Date | Date de sortie du jeu | Pas de valeurs manquantes |
| Price | Float | Prix initial du jeu | Pas de valeurs manquantes |
| MetacriticScore | Integer | Evaluation moyenne des critiques | Peut être 0 |
| Positive | Integer | Nombre de votes positifs | Peut être 0 |
| Negative | Integer | Nombre de votes négatifs | Peut être 0 |
| AveragePlaytimeForever | Integer | Temps de jeu moyen | Temps en minutes |
| Developers | String | Développeurs du jeu | - |

Table 2.2: Steam Charts on Steam (52287 x 5)

| Nom.colonne | Type | Signification | Caractéristique |
|---|---|---|---|
| AppID | Integer | Identifiant unique du jeu sur Steam | Clé primaire, unique |
| Month | String | Mois où les statistiques sont prises | Format de date |
| AvgPlayers | Float | Nombre moyen de joueurs actifs | Peut être 0 |
| Gain | Float | Gain du nombre de joueurs par rapport au mois précédent | - |
| PercentGain | Float | Pourcentage du gain du nombre de joueurs par rapport au mois précédent | - |
| PeakPlayers | Integer | Nombre maximal de joueurs actifs | Peut être 0 |

## 2.4 Les modèles conceptuels

### *2.4.1 Modèle MCD*

Le modèle conceptuel de données a été élaboré pour visualiser les relations entre nos différentes tables et assurer une structure logique de notre base de données. Le MCD, conçu à l'aide de l'outil en ligne Mocodo (https://www.mocodo.net/), montre comment les jeux vidéo, leurs prix, leurs évaluations, et données de joueurs actifs sont interconnectés, facilitant ainsi les analyses croisées nécessaires pour répondre à notre question de recherche.

- MCD réalisé avec le logiciel Mocodo :

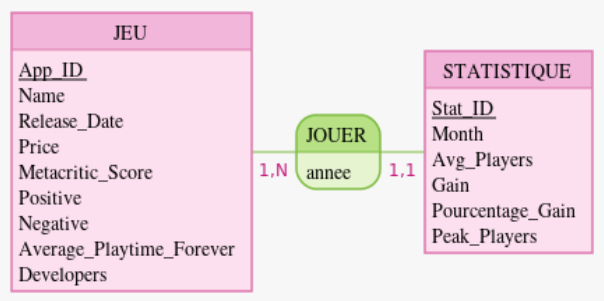



Figure 2.1: Relation de Jouabilité entre 'JEU' et 'STATISTIQUE' via l'association 'JOUER'.

### *2.4.2 Modèle MOD*

À partir de notre modèle conceptuel de données qui présente une relation déséquilibrée entre nos deux entités, nous avons créé un modèle relationnel avec le designer de phpMyAdmin (http://localhost/phpMyAdmin/), qui matérialise les concepts et les associations en une structure adaptée. Ce modèle est composé de deux tables principales et d'une associations. Les informations sur les jeux sont reliées aux statistiques de jeu correspondantes, avec l'identifiant de l'application (AppID) servant de clé étrangère pour relier les deux entités. La clé primaire de la seconde table est donc une clé composite, composée des variables "AppID" et "Month".

- MOD réalisé avec le designer de phpMyAdmin :

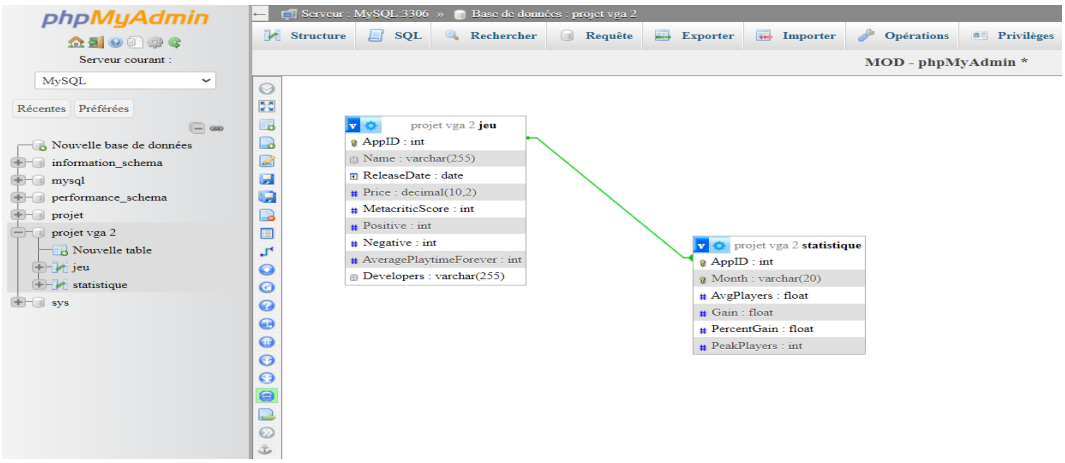



Figure 2.2: Modèle de Données des Jeux et Statistiques.

## 2.5 Requêtes réalisées

Dans un premier temps, nous avons exécuté des requêtes simples afin d'explorer différents angles de recherche. Ces démarches préliminaires nous ont permis de répondre à notre question de recherche grâce à des requêtes ultérieures, plus complexes mais également plus pertinentes. Les requêtes que nous considérons comme optimales sont les suivantes :

### *2.5.1 Requête 1 : Les moyennes*

Table 2.3: Requête 1, Les moyennes

| Moyenne des prix | Moyenne des notes Metacritic |
|---|---|
| 20.59 | 35.19 |

Cette requête sert à avoir la moyenne générale du prix et du ń MetacriticScore ż pour tous les jeux de notre base de données. Nous pourrions nous basé sur ces informartions pour créer des catégories afin de réaliser des statistiques descriptives.

### *2.5.2 Requête 2 : Les catégories des prix*

Table 2.4: Requête 2, Les catégories des prix

| Catégorie de Prix | Moyenne des joueurs actifs |
|---|---|
| Gratuit | 16146.02 |
| Inférieur à 20 | 2512.70 |
| Supérieur à 20 | 3625.50 |

Cette requête SQL est conçue pour segmenter les jeux vidéo en trois catégories de prix : gratuits, prix inférieurs à 20 euros et prix supérieurs à 20 euros. Pour chaque catégorie, la requête calcule la moyenne du nombre de joueurs actif (AvgPlayers).

En analysant ces résultats, on peut conclure que les jeux gratuits ont significativement plus de joueurs actifs en moyenne que les jeux payants, ce qui pourrait indiquer que l'absence de barrière financière à l'entrée est un facteur majeur d'attraction pour une plus grande base de joueurs. Cela est cohérent avec certaines tendances de l'industrie où les modèles free-to-play peuvent attirer de larges publics. Les jeux avec un prix supérieur à 20 euros ont également une base de joueurs actifs plus importante en moyenne que les jeux à moins de 20 euros, ce qui pourrait suggérer que les jeux à prix plus élevé possèdent soit une qualité ou une réputation qui attire les joueurs, soit qu'ils s'adressent à des niches de marché spécifiques prêtes à payer un premium pour des expériences meilleures. Cependant, ces résultats doivent être interprétés avec prudence, car d'autres facteurs (tels que la qualité du jeu, le marketing, la popularité de la franchise, le genre du jeu, etc.) peuvent également influencer le nombre de joueurs actifs et ne sont pas pris en compte dans notre analyse.

### *2.5.3 Requête 3 : Les developpeurs*

Table 2.5: Requête 3, Les developpeurs

| Développeur | Moyenne des notes Metacritic | Moyenne des gains de joueurs |
|---|---|---|
| ZA/UM | 97 | 39.95 |
| Beat Games | 93 | 18.07 |
| Relic Entertainment | 93 | 4.23 |
| Wube Software LTD. | 90 | 115.09 |
| Firaxis Games | 90 | 5.70 |
| ConcernedApe | 89 | 403.67 |
| Mega Crit Games | 89 | 136.00 |
| Motion Twin | 89 | 36.72 |
| SkyBox Labs,Big Huge Games | 89 | 1.19 |
| Cellar Door Games | 88 | 66.72 |

Pour identifier les meilleurs développeurs en fonction du score Metacritic et du gain de joueurs actifs, nous avons exécuté une requête SQL qui calcule la moyenne des scores Metacritic et des gains moyens des joueurs actifs pour les jeux de chaque développeur, puis ordonné les résultats pour mettre en évidence les meilleurs. En évaluant la moyenne du gain, nous pouvons déduire si un jeu réussit à élargir sa base de joueurs ou si, au contraire, il la perd sur la période étudiée.

Nous avons utilisé HAVING pour inclure uniquement les développeurs dont les jeux ont une moyenne de score Metacritic supérieure à 75 et un gain moyen de joueurs actifs positif, ce qui indique une performance favorable à la fois des critiques et en termes d'acquisition de joueurs. L'analyse montre des développeurs tels que ZA/UM et Beat Games avec des scores Metacritic très élevés et des gains significatifs de joueurs actifs, suggérant que leurs jeux sont non seulement bien reçus par les critiques mais attirent également de nouveaux joueurs. Par exemple, ZA/UM avec un score Metacritic moyen de 97 et un gain moyen de joueurs de près de 40 dénote un succès écrasant sur les deux fronts.

Il est également intéressant de noter que les développeurs bien connus et établis, tels que Beat Games et Relic Entertainment, apparaissent avec de bons scores Metacritic et des gains élevés de joueurs actifs. Cela peut indiquer que la notoriété de la marque et l'historique de développement de jeux de qualité contribuent à la croissance continue de la base de joueurs.

### *2.5.4 Requête 4 : Le Plus/Minus*

Table 2.6: Requête 4, Le Plus/Minus

| Jeu | Prix | Avis Metacritic | Mois | Moyenne des joueurs actifs | Moyenne des joueurs actifs de l'année dernière | +/- |
|---|---|---|---|---|---|---|
| Counter-Strike: Global Offensive | 0 | 83 | April 2020 | 857604.2 | 351989.9 | 505614.3 |
| Grand Theft Auto V | 0 | 96 | April 2016 | 31671.3 | 192714.0 | -161042.7 |

La requête SQL a pour but d'analyser les variations du nombre de joueurs actifs au cours des mois d'avril (mois choisi arbitrairement) pour des jeux sélectionnés, basée sur des critères précis. La requête inclut des colonnes clés qui sont critiques pour l'analyse :

1. AppID et Name pour identifier de manière unique et nommer le jeu.
2. Price pour observer l'influence du coût sur la participation des joueurs.
3. MetacriticScore pour filtrer les jeux selon leur accueil critique, ne retenant que ceux ayant un score de 70 ou plus.
4. AvgPlayers pour le nombre moyen de joueurs en avril de l'année courante, et
5. LastYearPlayers pour le nombre moyen de joueurs en avril de l'année précédente.

Le calcul du PlusMinus est au coeur de la requête, elle estla différence entre le nombre moyen de joueurs du mois d'avril courant (AvgPlayers) et celui de l'année précédente (LastYearPlayers). Ce calcul révèle les variations annuelles du nombre de joueurs, offrant un indicateur direct de la croissance ou de la réduction de l'engagement des joueurs. Il y a aussi un filtrage suplémentaire, seuls les jeux avec un score Metacritic d'au moins 70 sont inclus, focalisant l'analyse sur des jeux bien reçus par la critique. En plus, du fait que seuls les mois d'avril sont considérés, ce qui permet une comparaison cohérente d'une année sur l'autre.

Cette condition filtre le bruit potentiel que pourraient introduire des jeux moins performants ou mal reçus, permettant une évaluation plus précise de l'impact des critiques positives sur le succès d'un jeu. Les variations dans le nombre de joueurs pour ces jeux bien notés peuvent souvent refléter une réponse directe à des facteurs externes comme les campagnes promotionnelles, les mises à jour de contenu significatives ou les changements dans les tendances de consommation des médias.

Les résultats de cette analyse peuvent aider les éditeurs de jeux et les développeurs à optimiser leurs stratégies de lancement et de promotion. Par exemple, un jeu affichant une augmentation significative des joueurs pourrait indiquer une réception très favorable à une récente mise à jour ou à une modification de prix, tandis qu'une forte baisse pourrait signaler des problèmes nécessitant une attention rapide, tels que des problèmes techniques ou un contenu insatisfaisant.

Les insights tirés de cette analyse peuvent également guider les décisions de planification à long terme. Comprendre les tendances sur plusieurs années permet aux entreprises de mieux prédire les périodes de haute activité et d'ajuster leurs ressources en conséquence, que ce soit pour le support technique, le marketing ou le développement de contenu supplémentaire. En outre, ces données fournissent une fenêtre sur l'engagement des joueurs,

offrant une mesure claire de la rétention et de l'acquisition de joueurs au fil du temps. Ceci est particulièrement pertinent dans l'industrie du jeu vidéo où l'engagement des joueurs est un indicateur clé de succès continu. Les jeux qui réussissent à maintenir ou à augmenter leur base de joueurs sur des périodes prolongées peuvent souvent bénéficier de meilleures opportunités de monétisation et d'une communauté plus solide.

Cette requête SQL offre des insights utiles sur l'évolution annuelle du nombre de joueurs pour des jeux bien notés et peut partiellement répondre à votre question sur l'influence des critiques et du prix initial. Cependant, elle ne cible pas spécifiquement les premiers mois après la sortie des jeux, ni n'analyse directement le lien entre le prix initial et l'engagement des joueurs.

### *2.5.5 Requête 5 : Le Game Count*

Table 2.7: Requête 5, Le Gamecount

| Catégorie du jeu | Nombre de jeu | Moyenne des joueurs actifs |
|---|---|---|
| Note haute et bas prix | 201 | 9001.76 |
| Autre | 562 | 5344.39 |
| Note basse et haut prix | 187 | 3075.04 |

Cette requête permet de classé chaque jeu en fonction de son score MetacriticScore et de son prix. Les jeux avec un score Metacritic supérieur ou égal à 70 et un prix inférieur ou égal à 20 euros sont classés comme 'Note haute et bas prix'. Les jeux avec un score Metacritic inférieur à 70 et un prix supérieur à 20 euros sont classés comme 'Note basse et haut prix'. Tous les autres jeux sont classés comme 'Autre'. Pour chaque jeu, on calcule la moyenne des joueurs actifs. Elle permet aussi de regrouper les jeux par AppID, ainsi que par leur score Metacritic et leur prix, ce qui est essentiel pour obtenir une moyenne précise des joueurs actifs pour chaque jeu unique. La requête compte le nombre de jeux dans chaque catégorie et calcule la moyenne des joueurs actifs par catégorie, en arrondissant cette moyenne à deux décimales.

La pertinence de cette requête réside dans sa capacité à fournir des insights sur la popularité actuelle d'un jeu : En calculant le nombre moyen de joueurs actifs, nous pouvons évaluer quels jeux captent et maintiennent l'intérêt des joueurs sur la plateforme Steam. Cette information est cruciale pour comprendre si un prix plus bas et des critiques plus positives sont associés à une base de joueurs plus grande et plus stable, ce qui est une donnée clé pour la stratégie d'un prix et de marketing des éditeurs de jeux vidéo.

La catégorie 'Note haute et bas prix' a le plus grand nombre de joueurs actifs en moyenne, ce qui suggère que les jeux qui sont bien notés et qui sont offerts à un prix inférieur ou égal à 20 euros ont tendance à attirer et à maintenir une base de joueurs plus importante.

Les résultats suggèrent qu'un prix compétitif est important, surtout si le jeu a reçu des évaluations positives. Les producteurs peuvent envisager des stratégies de prix flexibles pour maximiser la base de joueurs actifs.

Les jeux avec des scores Metacritic élevés tendent à attirer plus de joueurs, ce qui met en lumière l'importance de viser une haute qualité dans le développement pour obtenir de bonnes critiques. Il semble y avoir un segment de marché distinct pour les jeux bien notés et bon marché qui pourraient représenter un bon retour sur investissement en termes de base de joueurs actifs.

L'analyse des jeux dans la catégorie 'Autre' pourrait révéler des niches ou des opportunités pour des jeux qui ne correspondent pas aux deux principaux ensembles de critères mais qui ont quand même une base de joueurs active significative.

En somme, pour les futurs développeurs ou producteurs de jeux, ces informations sont cruciales car elles offrent des insights sur ce qui peut influencer la popularité et la longévité d'un jeu sur le marché. En se basant sur de telles données, ils peuvent affiner leurs stratégies de développement, de marketing et de tarification pour améliorer les chances de succès commercial de leurs jeux.

# Chapitre 3

# Matériel et Méthodes

## 3.1 Logiciels

Au cours de ce projet nous avons utiliser de nombreux logiciels :

### *3.1.1 Les logiciels de communication et de partage des données*

1. WhatsApp [1]: Pour planifer notre projet, nous avons commmencer par utiliser ce service de discussion instantanée.
2. Google Docs/Google Drive [2]: Nous avons par la suite essayer de nous partager nos avancées avec ce logiciel mais nous nous sommes vite rendus compte qu'il existait de meilleure option.
3. GitHub [3]: Nous avons ensuite choisi de continuer avec GitHub, le partage de code se faisant plus simplement.

### *3.1.2 Les logiciels de rédaction*

1. Overleaf [4]: Lors du début du projet, nous avions commencé à rédiger notre rapport sur Overleaf comme Mme.Bringay l'avait conseillé lors de ses cours. Overleaf nous a aussi permit de corriger notre texte.
2. Rstudio [5]: À la fin de la préparation du projet, nous avions toujours pas rédiger le projet en .rmd mais nous sommes arrivés tout de même à le rendre dans les délais en utilisant la rédaction du rapport en tex d'Overleaf. Tout en ajoutant de nouvelles choses comme le fait d'utiliser directement les bases de données de PhpMyAdmin dans le rapport en .rmd comme Mme.Demangeot nous l'avais recommandé.
3. ChatGPT [6]: Son utilisation nous a facilité la rédaction de code sur RStudio lors du nettoyage des données mais aussi lors de ce rapport. Cela a permit de nous débloquer de situation où nous étions limité en terme de connaissance.
4. LanguageTool [7]: Cela nous a permis de corriger les éventuelles fautes d'orthographes et de syntaxes dans notre projet.

[1] Version : 2.24.8.86.
[2] Version 2.20.181.04.45.
[3] Service en ligne, mis à jour continuellement.
[4] Service en ligne, mis à jour continuellement.
[5] RStudio : 2023.12.1.
[6] Mise à jour du 13 février, 2024.
[7] Service en ligne, mis à jour continuellement.

### *3.1.3 Les logiciels de traitement et de gestion de données*

1. Excel/LibreOffice Calc [8]: Nous avons visualisé les données avec ces logiciels et observé de nombreux problèmes de formatage.
2. Mocodo [9]: Cela nous a servi pour la modélisation conceptuelle de données.
3. Wamp/PhpMyAdmin [10]: L'importation des données sur PhpMyAdmin permet la gestion des bases de données MySQL.
4. RStudio [11]: Nous a permis de nettoyer nos données et sélectionner les variables utiles.

## 3.2 Ordinateur

1. VivoBook ASUS X515JA, Windows, Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz, Système dexploitation 64 bits
2. Asus ROG FX503, Windows, Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, Système dexploitation 64 bits

[8] Microsoft Excel : Version 2403; LibreOffice Calc : Version 7.4.3.
[9] Version 4.2.4.
[10] Wampserver : 3.3.2; PhpMyAdmin : 5.2.1.
[11] RStudio : 2023.12.1.

# Chapitre 4

# Analyse Exploratoire des Données

L'analyse exploratoire des données constitue une étape cruciale de notre étude. Elle nous permet de comprendre les caractéristiques fondamentales des données et d'identifier les tendances, anomalies ou patterns éventuels. Cette compréhension initiale est essentielle pour orienter nos analyses statistiques ultérieures et pour assurer l'application correcte des méthodes statistiques.

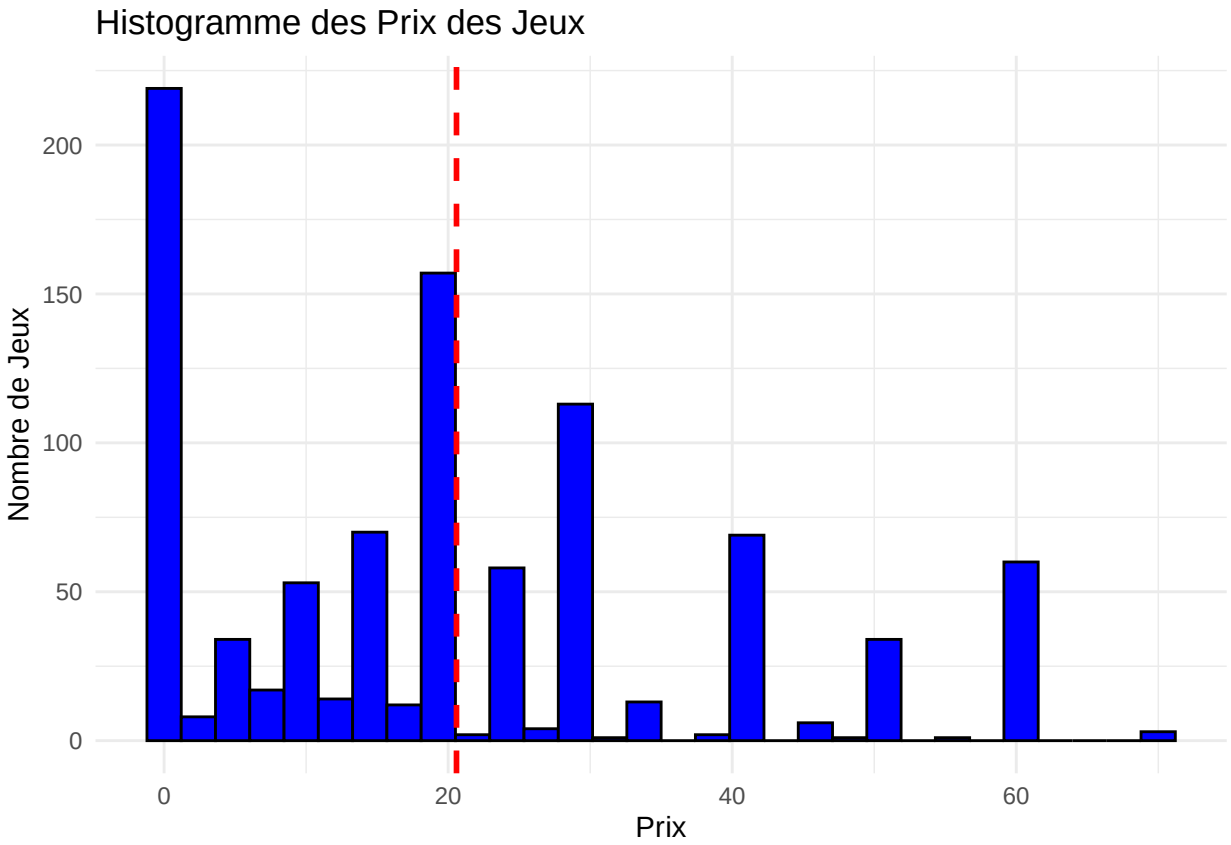
## 4.1 Modélisation de la variable 'Price'

se trouve la méthode utilisée pour calculer les statistiques est robuste et efficace pour notre analyse.

Table 4.1: Résumé des statistiques des prix des jeux

| Statistique | Valeur |
|---|---|
| Moyenne | 20.587 |
| Médiane | 19.990 |
| 1er Qu. | 4.490 |
| 3ème Qu. | 29.990 |
| Min. | 0.000 |
| Max. | 69.990 |
| Mode | 0.000 |
| Variance | 307.458 |
| Ecart-type | 17.534 |
| Coefficient de variation | 85.170 |
| Skewness | 0.681 |
| Kurtosis | 2.749 |

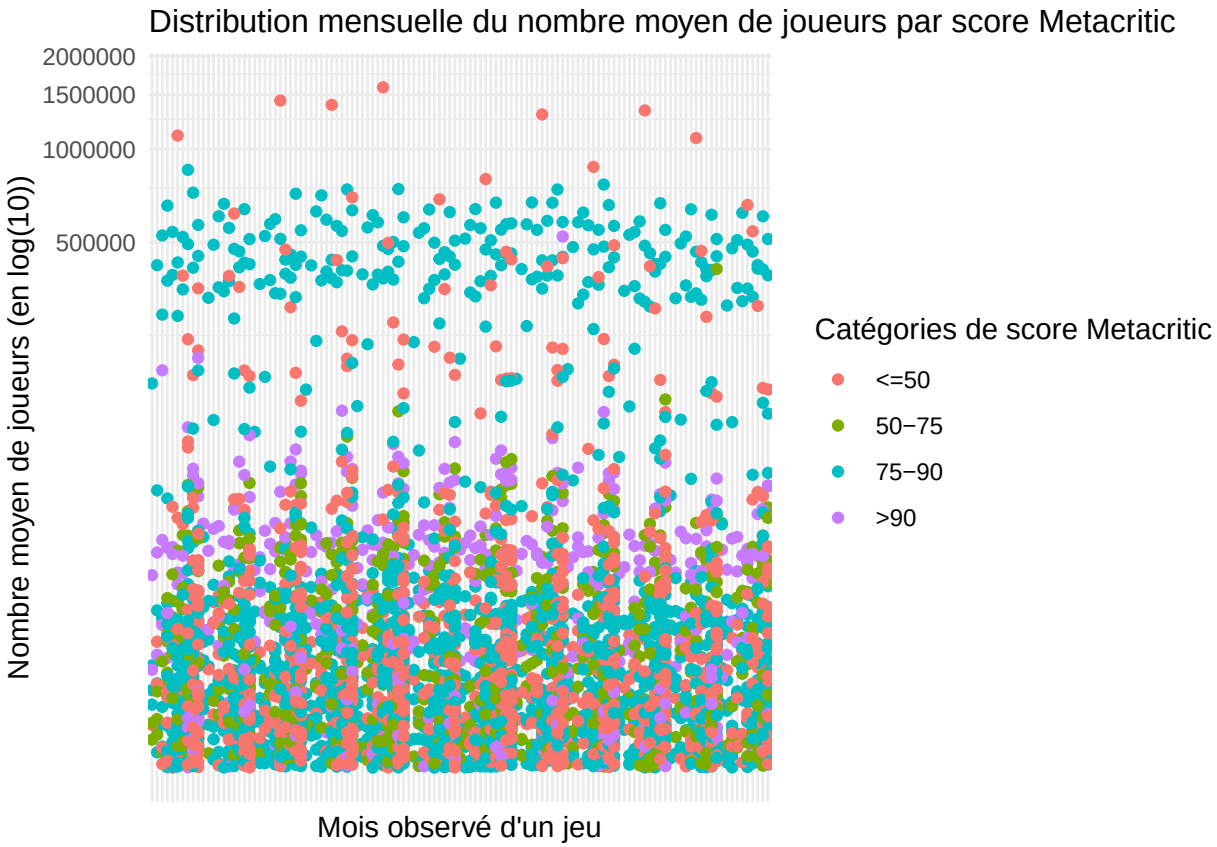
Pour visualiser la distribution des prix des jeux sur Steam, nous avons opté pour un histogramme, qui est particulièrement adapté pour montrer la fréquence des différentes gammes de prix dans notre jeu de données. Ce graphique nous permet d'observer la concentration des prix autour de certaines valeurs et d'identifier la présence de jeux gratuits, qui sont représentés comme une modalité distincte.




Cet histogramme montre que la majorité des jeux sont concentrés dans une gamme de prix inférieure à 20, avec une proportion significative de jeux proposés gratuitement. Précisément, 22.7% des jeux listés sont gratuits, soulignant la popularité des modèles freemium ou des offres promotionnelles initiales. En outre, environ 61.5% des jeux coûtent moins de 20, ce qui reflète une stratégie de tarification compétitive courante dans l'industrie pour attirer une base de joueurs plus large.

Les pics de distribution à 0 et autour de 10 à 20 peuvent être attribués à la popularité des jeux mobiles et indépendants, souvent tarifés dans cette fourchette pour maximiser l'accessibilité tout en garantissant une certaine rentabilité. Les jeux tarifés au-delà de 20, qui représentent une plus petite fraction du marché, sont généralement des titres plus substantiels, offrant des expériences plus profondes ou des contenus plus étendus, justifiant ainsi leur coût supérieur.

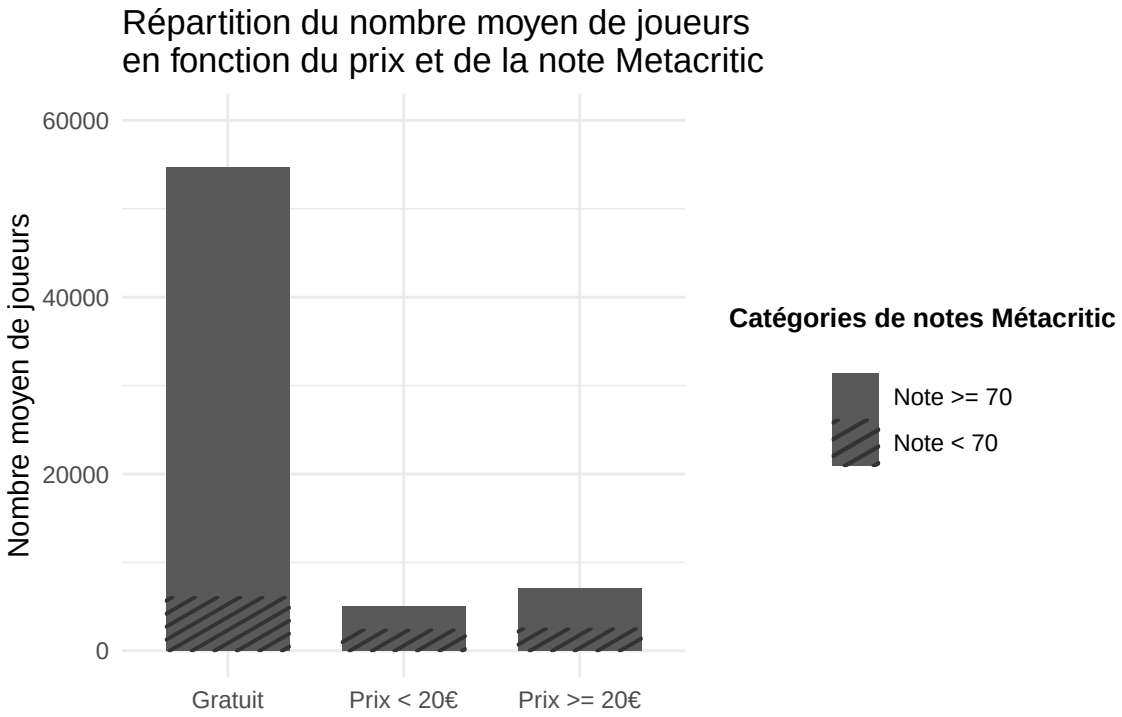
Cette distribution des prix a des implications directes tant pour les développeurs que pour les consommateurs, influençant les décisions d'achat des joueurs et les stratégies de tarification des éditeurs. Pour les développeurs, comprendre ce schéma peut aider à positionner leurs jeux de manière optimale dans un marché compétitif. Pour les consommateurs, cela signifie une diversité de choix qui peut s'adapter à divers budgets et préférences de jeu.




L'analyse de ce nuage de points révèle que la majorité des jeux ayant un score de presse supérieur à 90 ou un score situé entre 50 et 75 présentent un nombre moyen de joueurs par mois relativement faible, à l'exception de quelques titres. En revanche, pour les jeux ayant un score inférieur à 50, on observe une plus grande variabilité dans le nombre moyen de joueurs par mois, bien que la plupart de ces jeux aient un nombre moyen de joueurs/mois faible, certains d'entre eux enregistrent un nombre relativement important, voire très important pour une dizaine de jeux, se situant entre 250 000 et 1500 000.

L'échelle logarithmique en base 10, utilisée ici pour le nombre moyen de joueurs par mois, permet de mieux discerner les différences entre les jeux ayant de petits et de grands nombres de joueurs. Cette mise à l'échelle est particulièrement utile pour ce jeu de données, car sans elle, les quelques jeux avec des nombres extrêmement élevés de joueurs pourraient éclipser la variabilité parmi les jeux moins populaires. Grâce à l'échelle logarithmique, on peut observer que même parmi les jeux avec des scores de presse moins élevés, certains peuvent quand même attirer un nombre substantiel de joueurs, ce qui pourrait être masqué avec une échelle linéaire standard.

En ce qui concerne les jeux ayant un score Metacritic compris entre 75 et 90, on remarque que leur nombre moyen de joueurs est généralement plus élevé que dans les autres classes, le nombre moyen de joueurs pour ces jeux se situe majoritairement entre 200 000 et 750 000, démontrant ainsi une popularité plus marquée auprès des joueurs.




L'analyse du diagramme en barres met en évidence des disparités dans le nombre moyen de joueurs selon les catégories de jeux. Il apparaît clairement que les jeux gratuits ayant un score de presse supérieur à 70 bénéficient d'un nombre moyen de joueurs considérablement plus élevé par rapport aux autres catégories, bien que les jeux gratuits avec un score de presse inférieur à 70 enregistrent également un nombre important de joueurs, celui-ci reste inférieur à la première catégorie.

En ce qui concerne les jeux payants, on observe que ceux ayant un score Metacritic supérieur à 70 enregistrent un nombre moyen de joueurs plus important que les jeux dont le score est inférieur à 70. Cette tendance suggère que les scores attribués par la presse peuvent avoir un impact sur le nombre de joueurs moyen, en particulier pour les jeux payants.

En outre, les jeux gratuits et les jeux payants bien notés par la presse semblent être les catégories les plus susceptibles d'attirer un nombre important de joueurs.

# Chapitre 5

# Analyse et Résultats

## 5.1 Importance des modèles simples

Comme mentionné dans les consignes de projet de ce chapitre, il est crucial de commencer avec un modèle simple avant d'adopter des approches plus complexes. Cette méthode permet une meilleure compréhension des données et se révèle être davantage abordable pour notre niveau d'étude. Le principe d'utiliser un modèle que l'on maîtrise assure une interprétation correcte et fiable des résultats, facilitant ainsi les décisions basées sur ces analyses.

## 5.2 Test ANOVA

Pour aller plus loin dans notre analyse statistique, il est essentiel de comparer les moyennes des nombres de joueurs parmi les différentes catégories de jeux que nous nous sommes crée. Pour se faire, nous avons réalisé un test ANOVA (Analyse de Variance) sur les données des jeux à notre disposition. Nous avons regroupé les jeux en fonction de leur identifiant unique (AppID) et avons introduit une nouvelle variable appelée "Catégorie". Cette variable indique si un jeu appartient à l'une des six catégories que nous avons définies :

1. Jeux gratuits avec un score inférieur à 70,
2. Jeux gratuits avec un score supérieur à 70,
3. Jeux payants dont le prix est inférieur à 20 avec un score inférieur à 70,
4. Jeux payants dont le prix est inférieur à 20 avec un score supérieur à 70,
5. Jeux payants dont le prix est supérieur à 20 avec un score inférieur à 70,
6. Jeux payants dont le prix est supérieur à 20 avec un score supérieur à 70.

Le but de cette approche est de déterminer s'il existe une différence significative dans le nombre moyen de joueurs entre les différentes catégories de jeux. Pour effectuer ces tests ANOVA sur R, il est recommandé d'utiliser la fonction aov().

## 5.3 Résultats

Le test ANOVA réalisé pour comparer les moyennes du nombre de joueurs actifs entre différentes catégories de prix et de scores Metacritic a montré des différences significatives. Comme indiqué dans la partie statistique descriptive, nous observons que les jeux gratuits ou à bas prix ayant une note élevée attirent et maintiennent un nombre plus élevé de joueurs, ce qui est démontré par les résultats statistiquement significatifs avec une valeur de Pr(>F) inférieure à 2e-16 (16((2 * 10  -16))).

Ces résultats confirment l'hypothèse que le prix et la qualité perçue, mesurée par les scores Metacriticscore, jouent un rôle crucial dans la popularité des jeux sur Steam. Ces variables ont un impact significatif sur le nombre de joueurs, soulignant l'importance pour les développeurs de cibler ces aspects lors de la mise sur le marché de nouveaux jeux.

Les analyses ont montré que des tests statistiques plus avancés tels que l'ANOVA peuvent fournir des éléments précieux pour notre étude.

Les implications de ces résultats sont significatives pour la stratégie de développement et de marketing des jeux. Les développeurs doivent considérer le positionnement de prix soigneusement tout en s'assurant que la qualité du jeu est susceptible de recevoir des critiques positives pour maximiser l'attraction et la rétention des joueurs. La confirmation de ces effets par le test ANOVA renforce la nécessité d'une approche équilibrée entre prix attractif et haute qualité perçue pour réussir dans un marché aussi compétitif que celui des jeux sur Steam.

## 5.4 Conclusion

Ce chapitre a mis en lumière l'importance d'utiliser des modèles comme celui de l'ANOVA, pour montrer comment des variables comme le prix et les critiques impactent significativement le nombre de joueurs actifs au sein d'une communanté. Ces découvertes doivent être intégrées dans les phases de planification et de développement des jeux pour optimiser leur succès commercial.

Il serait judicieux d'explorer davantage comment ces facteurs interagissent avec d'autres variables telles que le genre du jeu, les campagnes tarifaires promotionnelles, ou les mises à jour post-lancement. Des analyses supplémentaires pourraient inclure des modèles de régression multiple ou des techniques de machine learning pour prédire plus précisément les tendances du marché et les comportements des joueurs, permettant ainsi une adaptation plus fine des stratégies de lancement et de promotion des jeux.

# Chapitre 6

# Discussion

Les résultats du test d'ANOVA ont mis en évidence que la variable "catégorie" avait un effet hautement significatif sur le nombre moyen de joueurs, avec une valeur de Pr(>F) (qui représente la valeur p associée à la statistique F et qui indique la significativité des différences entre les groupes ou l'effet d'un facteur) inférieure à 2e-16((2 * 10 ^ -16). Cette valeur étant bien inférieure au seuil de significativité de 0,05, cela indique que les différences observées entre les catégories de jeux ne sont pas dues au hasard et que le prix et le score attribué par la presse à un jeu ont un impact important sur le nombre moyen des joueurs par mois.

## 6.1 Synthèse des résultats

Dans le chapitre précédent, nous avons démontré à travers un test ANOVA que les catégories de prix et les scores Metacritic ont un effet significatif sur le nombre moyen de joueurs actifs. Les résultats obtenus sont hautement significatifs, avec une valeur de Pr(>F) inférieure à 2e-16, indiquant une influence claire de ces facteurs sur la popularité des jeux sur Steam (Chapitre 5, section 5.3). Cette découverte est en adéquation avec les observations préliminaires faites lors de l'analyse exploratoire des données où il a été observé que les jeux gratuits ou à bas prix et bien notés attirent et maintiennent un nombre plus élevé de joueurs (Chapitre 4).

## 6.2 Impact des prix et de MetacriticScore

Les données indiquent que non seulement les jeux gratuits mais aussi ceux vendus à un prix inférieur à 20 euros, s'ils sont accompagnés de critiques positives, ont tendance à attirer un nombre plus élevé de joueurs. Cette tendance souligne l'importance d'une tarification stratégique combinée à la qualité perçue du jeu, comme l'ont montré les catégories de jeux dans nos analyses, où les jeux avec un score Metacritic supérieur à 70 enregistrent systématiquement un plus grand nombre de joueurs, quelle que soit la tranche de prix (Chapitre 4, Tableau des répartitions par catégorie de prix et note Metacritic).

Cette dynamique est également visible dans le comportement des joueurs, où les jeux ayant reçu des critiques favorables attirent non seulement des joueurs initialement mais parviennent aussi à les retenir, ce qui suggère une corrélation entre les scores de critiques et la durabilité de l'engagement des joueurs. La discussion de ces résultats dans la section des analyses exploratoires montre clairement cette tendance (Chapitre 4).

## 6.3 Conclusion

La capacité à interpréter correctement l'impact des prix et des critiques ouvre la voie à des décisions de développement et de marketing plus informées. En utilisant ces informations, les développeurs peuvent mieux positionner leurs jeux sur le marché, attirant efficacement et durablement les joueurs.

# Chapitre 7

# Conclusion et perspectives

## 7.1 Conclusions principales

Notre étude a établi que le prix et les critiques des jeux ont une influence significative sur le nombre de joueurs actifs sur Steam. Les jeux bien notés et à bas prix tendent à attirer et à retenir un plus grand nombre de joueurs, validant ainsi l'importance de ces deux facteurs pour les développeurs et les marketeurs de jeux vidéo.

## 7.2 Considérations pour les développeurs de jeux vidéo

Les développeurs doivent prendre en compte ces résultats pour optimiser leurs stratégies de lancement. Placer un jeu à un prix compétitif tout en s'assurant de la qualité susceptible d'attirer des critiques positives peut être une stratégie gagnante. Les jeux bien évalués par les critiques semblent bénéficier d'une base de joueurs plus large et plus durable, ce qui est crucial pour la rentabilité à long terme au sein de la communauté de Steam.

## 7.3 Perspectives

A court terme, il serait pertinent d'approfondir l'analyse pour explorer les effets des différentes informations que possèdent les jeux et leur impact sur l'engagement des joueurs. Sur le long terme, il serait judicieux d'étudier l'impact des nouvelles technologies de jeu et des modèles économiques émergents, tels que les jeux basés sur le cloud et les abonnements, pour anticiper et influencer les tendances futures du marché. L'intégration de ces perspectives dans la planification stratégique et opérationnelle pourrait potentiellement transformer les pratiques de développement de jeux, maximisant ainsi l'impact et le succès des jeux sur des plateformes comme Steam.

## 7.4 Difficultés rencontrées

Partie Base de Données : Le principale problème que nous avons rencontré réside dans la recherche de jeu de données adéquats à notre question de recherche. Nous avons changé de datasets plus de quatre fois et modifié notre question de recherche à plusieurs reprises à cause de la difficulté à trouver des données adéquates, au cours de notre projet. Cette partie nous à sembler très limitante pour notre créativité. Le partage des bases de données après le prétraitement notamment, a posé problème, en particulier avant de choisir GitHub, qui s'est révélé peu intuitif initialement. Partie Statistique : Nous n'avons pas rencontré de difficultés majeures pour le nettoyage des données, car le dataset était déjà bien structuré. Cependant, nous avons dû effectuer un tri minutieux des noms des jeux pour éliminer les écritures en chinois simplifié et convertir les dates en valeurs numériques, ce qui a pris beaucoup de temps. Cela aurait pu etre très endommageant pour l'importation de nos dataset sur le serveur de phpMyAdmin et effectuer la partie requete SQL. L'application des notions de cours, notamment le codage en R Markdown,

a été un défi, mais notre dynamisme dans l'apprentissage nous a permis de surmonter ces obstacles.

## 7.5 Synthèse et engagement du groupe

Malgré les difficultés, l'engagement de chaque membre du groupe a été essentiel pour surmonter les défis et produire un rapport final qui reflète notre vision et répond à nos objectifs académiques. Cette expérience a renforcé notre capacité de collaboration et d'utilisation de diverses ressources pédagogiques, y compris des outils en ligne comme ChatGPT, pour soutenir notre processus d'apprentissage.

En conclusion, ce projet a été une opportunité précieuse pour développer nos compétences pratiques et théoriques en science des données.

# Webographie


1. OpenAI Blog sur ChatGPT - Un aperçu détaillé de ChatGPT, un modèle de langage avancé développé par OpenAI. URL: https://openai.com/blog/chatgpt
2. SpringeR - Livre R - Un guide complet sur le logiciel statistique R, disponible sous forme de PDF. URL: https://biostatisticien.eu/springeR/livreR.pdf
3. Kaggle Datasets - Une plateforme offrant un large éventail de datasets pour les projets de science des données. URL: https://www.kaggle.com/datasets
4. LanguageTool - Un outil en ligne pour la correction grammaticale et orthographique. URL: https://languagetool.org/fr
5. Overleaf Project - Une plateforme de rédaction et de collaboration en LaTeX. URL: https://www.overleaf.com/project
6. Mocodo - Outil en ligne pour la modélisation conceptuelle de données. URL: https://www.mocodo.net/
7. phpMyAdmin (Localhost) - Interface de gestion pour les bases de données MySQL, généralement utilisée localement. URL: http://localhost/phpmyadmin/
8. datanovia - Outils en ligne pour aide statistique descriptive URL: https://www.datanovia.com/

# Annexes

## Codes

*Code du nettoyage des données*

```
# Charger les bibliothèques
library(dplyr)
# Utilisée pour les manipulations de données.
library(stringr)
# Permet de travailler facilement avec des chaînes de caractères.
library(lubridate)
```

```
## Warning: le package 'lubridate' a été compilé avec la version R 4.3.3
```

```
# Simplifie la gestion des dates.

# Lire les fichiers
# On charge les données depuis les fichiers CSV sur le disque local.
games <- read.csv(
  "C:\\Users\\antoc\\OneDrive\\Bureau\\Projet BD-SDD2\\games.csv",
  stringsAsFactors = FALSE)
steam_charts <- read.csv(
  "C:\\Users\\antoc\\OneDrive\\Bureau\\Projet BD-SDD2\\steam_charts.csv",
  stringsAsFactors = FALSE)

# Renommer et préparer les colonnes
# Ici, on renomme certaines colonnes pour simplifier
# leur usage plus tard dans les scripts.
# On a utilisé ChatGPT pour le code qui utilise la fonction mutate
steam_charts <- steam_charts %>%
  rename(AppID = App.ID, Name = Game, `%Gain` = X..Gain) %>%
  mutate(Gain = replace(Gain, Gain == "-", 0),
         # Remplacer les tirets par des zéros pour éviter les erreurs de type.
         `%Gain` = replace(`%Gain`, `%Gain` == "-", 0)) %>%
  mutate(
    Gain = as.numeric(gsub("%", "", as.character(Gain), fixed = TRUE)),
    # Convertir les gains en numérique en supprimant le signe %.
    `%Gain` = as.numeric(gsub("%", "", as.character(`%Gain`), fixed = TRUE))
  )

# Filtrer les jeux
# On élimine les jeux en chinois simplifié et
# on ne garde que ceux présents dans steam_charts.
```

```r
# On a utilisé ChatGPT pour le codes qui modifie le format de la date

games_filtered <- games %>%
  rename(ReleaseDate = `Release.date`) %>%
  filter(AppID %in% steam_charts$AppID) %>%
  filter(Supported.languages != "['Simplified Chinese']") %>%
  mutate(ReleaseDate = format(mdy(ReleaseDate), "%Y/%m/%d"))
# Formatage de la date.

# Nettoyage supplémentaire pour aligner les jeux et les données Steam
steam_charts_filtered <- steam_charts %>%
  filter(AppID %in% games_filtered$AppID)

# Préparation finale des données de jeux
filtered_games_cleaned <- games_filtered %>%
  select(
    -Estimated.owners, -Peak.CCU, -Required.age, -DLC.count,
    -About.the.game,
    -Supported.languages, -Full.audio.languages, -Reviews, -Header.image,
    -Website,
    -Support.url, -Support.email, -Windows, -Mac, -Linux,
    -Metacritic.url, -User.score, -Score.rank, -Achievements,
    -Recommendations,
    -Notes, -Average.playtime.two.weeks, -Median.playtime.forever,
    -Median.playtime.two.weeks, -Publishers, -Categories,
    -Genres, -Tags,
    -Screenshots, -Movies
  ) %>%
  arrange(AppID)
# Tri des données par AppID pour une meilleure organisation.

# Sauvegarde des données nettoyées
write.table(filtered_games_cleaned,
"C:\\Users\\antoc\\OneDrive\\Bureau\\Projet BD-SDD2\\games_clean.csv",
    sep = ";", row.names = FALSE, col.names = TRUE,
    fileEncoding = "UTF-8", quote = TRUE)

# Préparation et sauvegarde finale des données Steam Charts
steam_charts_cleaned <- steam_charts_filtered %>%
  select(AppID, -Name, everything()) %>%
  arrange(AppID)

write.table(steam_charts_cleaned,
"C:\\Users\\antoc\\OneDrive\\Bureau\\Projet BD-SDD2\\steam_charts_clean.csv",
            sep = ";", row.names = FALSE, col.names = TRUE,
            fileEncoding = "UTF-8", quote = TRUE)
```

*Code de la requête 1*

```
query <- "
SELECT ROUND(AVG(Price), 2) as 'Price',
       ROUND(AVG(MetacriticScore), 2) as 'MetacriticScore'
FROM jeu;
"
results_moy <- dbGetQuery(con, query)
```

```
results_moy <- results_moy %>%
  rename(
    `Moyenne des prix` = Price,
    `Moyenne des notes Metacritic` = MetacriticScore
  )
kable(results_moy, "latex", booktabs = TRUE, longtable = TRUE,
      caption= "Requête 1, Les moyennes" ) %>%
  kable_styling(font_size = 10) %>%
  column_spec(1, width = "3cm")
```

Table 7.1: Requête 1, Les moyennes

| Moyenne des prix | Moyenne des notes Metacritic |
|---|---|
| 20.59 | 35.19 |

*Code de la requête 2*

```
query <- "
SELECT
  CASE
    WHEN j.Price = 0 THEN 'Gratuit'
    WHEN j.Price < 20 THEN 'Inférieur à 20'
    ELSE 'Supérieur à 20'
  END AS PriceCategory,
  ROUND(AVG(s.AvgPlayers), 2) as AvgActivePlayers
FROM jeu j
JOIN statistique s ON j.AppID = s.AppID
GROUP BY PriceCategory
ORDER BY CASE
           WHEN PriceCategory = 'Gratuit' THEN 1
           WHEN PriceCategory = 'Inférieur à 20' THEN 2
           ELSE 3
         END;
"
results_pri <- dbGetQuery(con, query)
```

```
results_pri <- results_pri %>%
  rename(
    "Catégorie de Prix" = PriceCategory,
    "Moyenne des joueurs actifs" = AvgActivePlayers
```

```
  )
kable(results_pri, "latex", booktabs = TRUE, longtable = TRUE,
      caption= "Requête 2, Les catégories des prix" ) %>%
  kable_styling(font_size = 10) %>%
  column_spec(1, width = "4cm")
```

Table 7.2: Requête 2, Les catégories des prix

| Catégorie de Prix | Moyenne des joueurs actifs |
|---|---|
| Gratuit | 16146.02 |
| Inférieur à 20 | 2512.70 |
| Supérieur à 20 | 3625.50 |

*Code de la requête 3*

Table 7.3: Requête 3, Les developpeurs

| Développeur | Moyenne des notes Metacritic | Moyenne des gains de joueurs |
|---|---|---|
| ZA/UM | 97 | 39.95 |
| Beat Games | 93 | 18.07 |
| Relic Entertainment | 93 | 4.23 |
| Wube Software LTD. | 90 | 115.09 |
| Firaxis Games | 90 | 5.70 |
| ConcernedApe | 89 | 403.67 |
| Mega Crit Games | 89 | 136.00 |
| Motion Twin | 89 | 36.72 |
| SkyBox Labs,Big Huge Games | 89 | 1.19 |
| Cellar Door Games | 88 | 66.72 |

*Code de la requête 4*

```
query <- "
WITH PlayerDifferences AS (
    SELECT
        j.AppID,
        j.Name,
        j.Price,
        j.MetacriticScore,
        s.Month,
        ROUND(s.AvgPlayers, 2) AS AvgPlayers,
        LAG(ROUND(s.AvgPlayers, 2)) OVER (
        PARTITION BY j.AppID ORDER BY s.Month) AS LastYearPlayers,
        ROUND((s.AvgPlayers - LAG(s.AvgPlayers) OVER (
        PARTITION BY j.AppID ORDER BY s.Month)), 2) AS PlusMinus
    FROM
        jeu j
    JOIN
        statistique s ON j.AppID = s.AppID
    WHERE
```

```
        s.Month LIKE 'April%' AND
        j.MetacriticScore >= 70
)

SELECT
    Name,
    Price,
    MetacriticScore,
    Month,
    AvgPlayers,
    LastYearPlayers,
    PlusMinus
FROM
    PlayerDifferences
WHERE
    PlusMinus = (SELECT MAX(PlusMinus) FROM PlayerDifferences)
    OR PlusMinus = (SELECT MIN(PlusMinus) FROM PlayerDifferences)
ORDER BY
    PlusMinus DESC;
"
results_plusminus <- dbGetQuery(con, query)
```

```
results_plusminus <- results_plusminus %>%
  rename(
    "Jeu" = Name,
    "Prix" = Price,
    "Avis Metacritic"= MetacriticScore,
    "Mois"= Month,
    "Moyenne des joueurs actifs" = AvgPlayers,
    "Moyenne des joueurs actifs de l'année dernière" = LastYearPlayers,
    "+/-" = PlusMinus
  )
kable(results_plusminus, "latex", booktabs = TRUE, longtable = TRUE,
      caption= "Requête 4, Le Plus/Minus" ) %>%
  kable_styling(font_size = 10) %>%
  column_spec(1, width = "2cm")%>%
  column_spec(3, width = "2cm")%>%
  column_spec(4, width = "0.5cm")%>%
  column_spec(5, width = "3cm")%>%
  column_spec(6, width = "3cm")
```

Table 7.4: Requête 4, Le Plus/Minus

| Jeu | Prix | Avis Metacritic | Mois | Moyenne des joueurs actifs | Moyenne des joueurs actifs de l'année dernière | +/- |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| Counter-Strike: Global Offensive | 0 | 83 | April 2020 | 857604.2 | 351989.9 | 505614.3 |
| Grand Theft Auto V | 0 | 96 | April 2016 | 31671.3 | 192714.0 | -161042.7 |

*Code de la requête 5*

```
query <- "
WITH ScoredGames AS (
  SELECT
    j.AppID,
    ROUND(AVG(s.AvgPlayers), 2) AS AvgActivePlayers,
    CASE
      WHEN j.MetacriticScore >= 70
        AND j.Price <= 20 THEN 'Note haute et bas prix'
      WHEN j.MetacriticScore < 70
        AND j.Price > 20 THEN 'Note basse et haut prix'
      ELSE 'Autre'
    END AS Category
  FROM jeu j
  INNER JOIN statistique s ON j.AppID = s.AppID
  GROUP BY j.AppID, j.MetacriticScore, j.Price
)
SELECT
  Category,
  COUNT(AppID) AS GameCount,
  ROUND(AVG(AvgActivePlayers), 2) AS AvgActivePlayers
FROM ScoredGames
GROUP BY Category;
"
results_gamecount <- dbGetQuery(con, query)
```

```
results_gamecount <- results_gamecount %>%
  rename(
    "Catégorie du jeu" = Category,
    "Nombre de jeu" = GameCount,
    "Moyenne des joueurs actifs"= AvgActivePlayers
  )
kable(results_gamecount, "latex", booktabs = TRUE, longtable = TRUE,
      caption= "Requête 5, Le Gamecount" ) %>%
  kable_styling(font_size = 10)
```

Table 7.5: Requête 5, Le Gamecount

| Catégorie du jeu | Nombre de jeu | Moyenne des joueurs actifs |
|---|---|---|
| Note haute et bas prix | 201 | 9001.76 |
| Autre | 562 | 5344.39 |

| | | |
|---|---|---|
| Note basse et haut prix | 187 | 3075.04 |

*Code du la modélisation de 'price'*

```
library(moments)
library(magrittr)

# Données
jeu <- dbReadTable(con, "jeu")
jeu_price <- jeu$Price

# Calcul des statistiques
table_price <- table(jeu_price)

mode_price <- as.numeric(names(which.max(table(jeu_price))))

variance_price <- round(var(jeu_price) * (length(jeu_price) - 1)
                        / length(jeu_price), 3)
ecart_type_price <- round(sqrt(variance_price), 3)
mean_price <- mean(jeu_price)
co_price <- round((ecart_type_price / mean_price) * 100, 3)
skewness_price <- round(skewness(jeu_price), 3)
kurtosis_price <- round(kurtosis(jeu_price), 3)
summary_price <- summary(jeu_price)

#Utilisation de Chat gpt pour la création du kable
# Extraction des valeurs pertinentes
mean_value <- round(summary_price["Mean"], 3)
median_value <- round(summary_price["Median"], 3)
q1_value <- round(summary_price["1st Qu."], 3)
q3_value <- round(summary_price["3rd Qu."], 3)
min_value <- round(summary_price["Min."], 3)
max_value <- round(summary_price["Max."], 3)

# Création du dataframe
result_df_ <- data.frame(Statistique = c("Moyenne", "Médiane", "1er Qu.",
                                         "3ème Qu.", "Min.", "Max.",
                                         "Mode","Variance",
                                         "Ecart-type",
                                         "Coefficient de variation",
                                         "Skewness", "Kurtosis"),
  Valeur = c(mean_value, median_value, q1_value, q3_value, min_value,
             max_value, mode_price, variance_price, ecart_type_price,
             co_price, skewness_price, kurtosis_price)
)
# Affichage du tableau
kable(result_df_, caption = "Résumé des statistiques des prix des jeux")
```

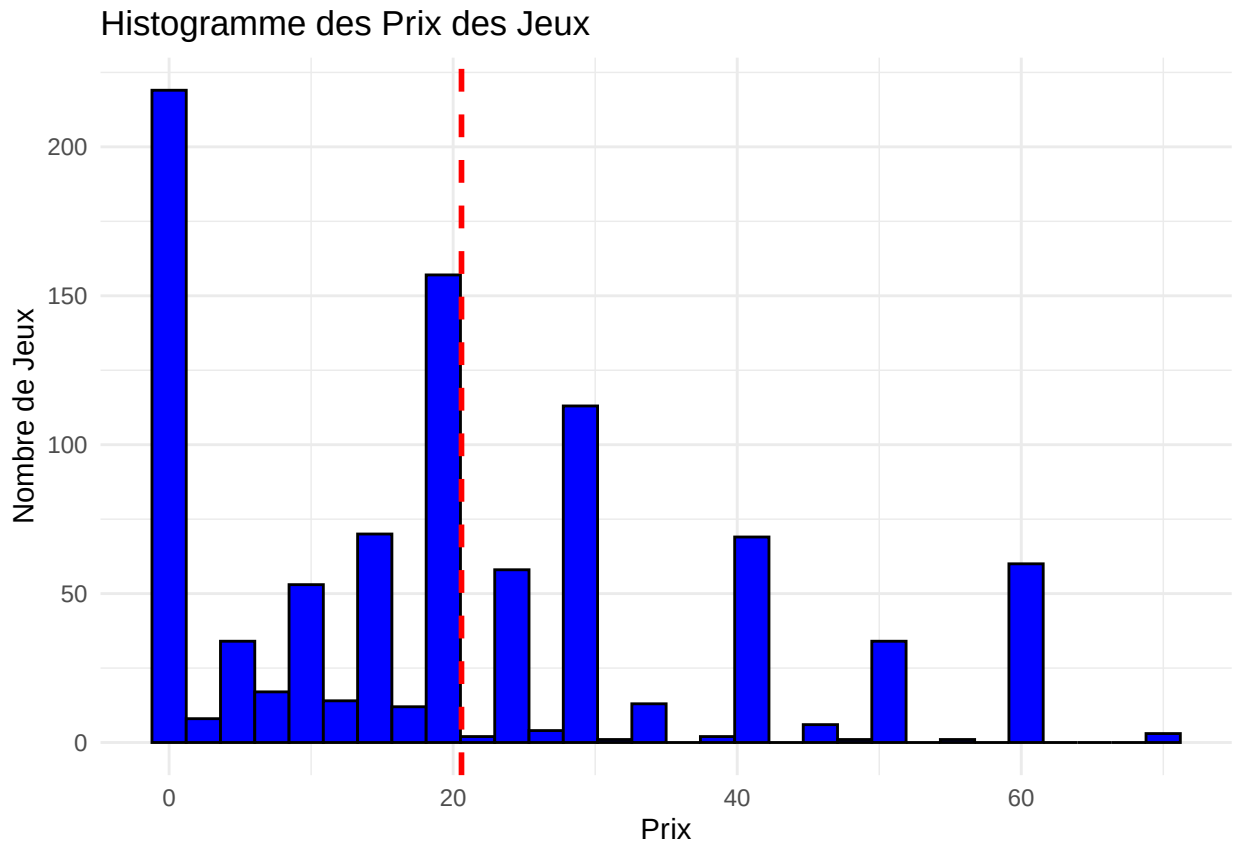
Table 7.6: Résumé des statistiques des prix des jeux

| Statistique | Valeur |
|---|---|
| Moyenne | 20.587 |
| Médiane | 19.990 |
| 1er Qu. | 4.490 |
| 3ème Qu. | 29.990 |
| Min. | 0.000 |
| Max. | 69.990 |
| Mode | 0.000 |
| Variance | 307.458 |
| Ecart-type | 17.534 |
| Coefficient de variation | 85.170 |
| Skewness | 0.681 |
| Kurtosis | 2.749 |

*Code de l'histogramme*

```
library(ggplot2)

jeu_price <- jeu$Price

# Calcul des statistiques descriptives
summary_stats <- data.frame(
  Statistique = c("Mean", "Median", "1st Qu.", "3rd Qu.", "Min.", "Max.",
                  "Mode", "Variance", "Ecart-type",
                  "Coefficient of Variation",
                  "Skewness", "Kurtosis"),
  Valeur = c(
    mean = mean(jeu_price, na.rm = TRUE),
    median = median(jeu_price, na.rm = TRUE),
    first_quartile = quantile(jeu_price, 0.25, na.rm = TRUE),
    third_quartile = quantile(jeu_price, 0.75, na.rm = TRUE),
    min = min(jeu_price, na.rm = TRUE),
    max = max(jeu_price, na.rm = TRUE),
    mode = as.numeric(names(which.max(table(jeu_price)))),
    variance = var(jeu_price, na.rm = TRUE),
    sd = sd(jeu_price, na.rm = TRUE),
    cv = (sd(jeu_price, na.rm = TRUE) /
            mean(jeu_price, na.rm = TRUE)) * 100,
    skewness = moments::skewness(jeu_price, na.rm = TRUE),
    kurtosis = moments::kurtosis(jeu_price, na.rm = TRUE) - 3
  )
)

# Créer un histogramme avec ggplot2
p <- ggplot(jeu, aes(x = Price)) +
  geom_histogram(bins = 30, fill = "blue", color = "black") +
```

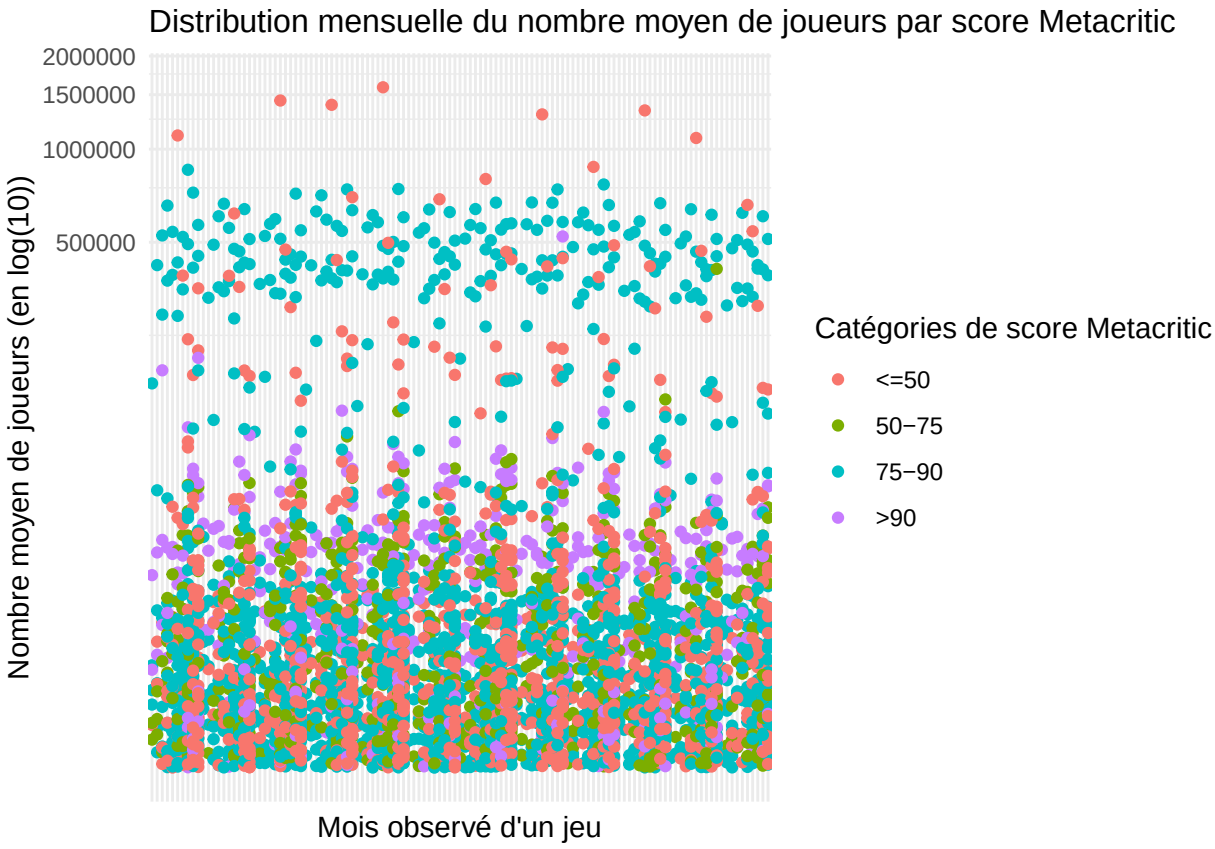
```r
  geom_vline(aes(xintercept = mean(Price, na.rm = TRUE)), color = "red",
             linetype = "dashed", size = 1) +
  labs(title = "Histogramme des Prix des Jeux", x = "Prix", y =
         "Nombre de Jeux") +
  theme_minimal()

# Afficher l'histogramme
print(p)
```




*Code du nuage de point*

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

statistique <- dbReadTable(con, "statistique")
# Fusion des dataframes 'statistique' et 'jeu' par 'AppID'
donnee_fusion <- merge(statistique, jeu, by = "AppID")

# Filtrer les données pour retirer les mois avec peu de joueurs
donnee_fusion <- donnee_fusion %>%
  filter(AvgPlayers > 10000)

# Création d'une variable catégorielle pour 'Metacritic.score'
donneesMetacriticScore_groupees <- cut(
  donnee_fusion$MetacriticScore,
```

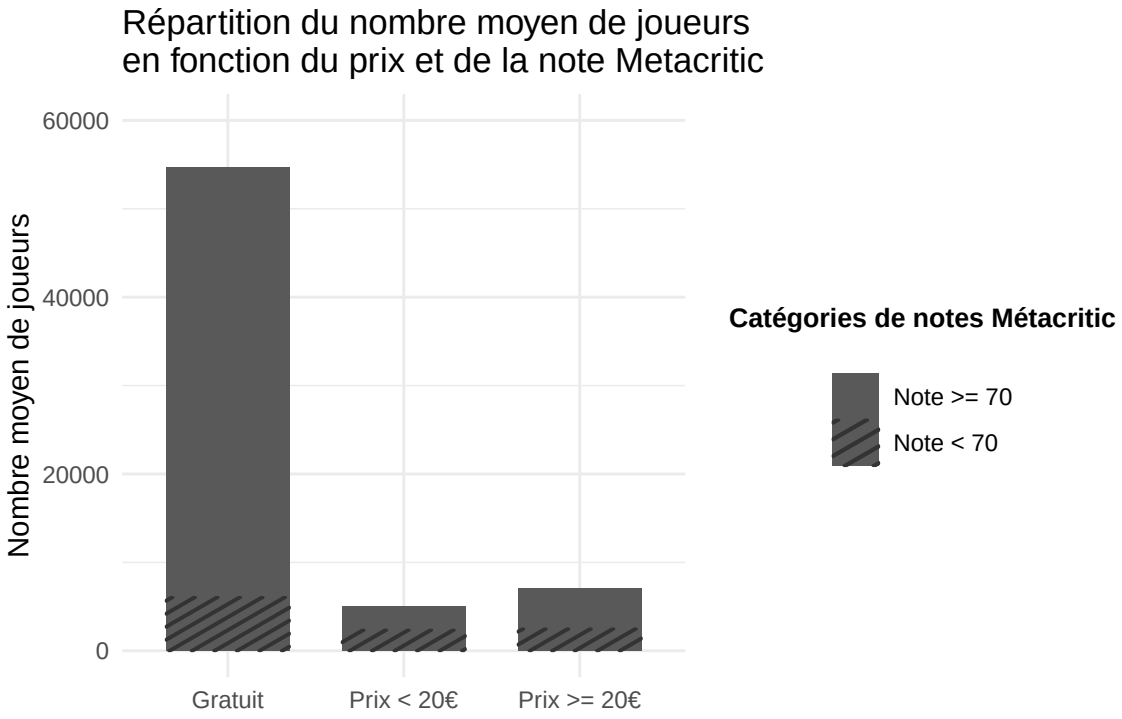
```r
  breaks = c(-Inf, 50, 75, 90, Inf),
  labels = c("<=50", "50-75", "75-90", ">90")
)

# Ajout de cette nouvelle variable catégorielle au dataframe
donnee_fusion$ScoreGroup <- donneesMetacriticScore_groupees

# Création du nuage de points avec ggplot2
ggplot(donnee_fusion, aes(x = Month, y = AvgPlayers, color = ScoreGroup)) +
  geom_point() +
  labs(
    x = "Mois observé d'un jeu",
    y = "Nombre moyen de joueurs (en log(10))",
    title = "Distribution mensuelle du nombre moyen de joueurs par score Metacritic",
    color = "Catégories de score Metacritic"
  ) + coord_trans(y ="log10")+
  theme_minimal() +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    plot.title = element_text(size = 12)
  )
```




```r
# coord_cartesian(ylim = c(1, 2000000))
```

*Code du diagramme*

```r
library(dplyr)
library(ggplot2)
library(ggpattern)

# Préparation des données
jeux_categorises <- jeu %>%
  inner_join(statistique, by = "AppID") %>%
  mutate(
    PriceCategory = case_when(
      Price == 0 ~ "Gratuit",
      Price > 0 & Price < 20 ~ "Prix < 20",
      Price >= 20 ~ "Prix >= 20"
    ),
    Pattern = ifelse(MetacriticScore < 70, "stripe", "none")
  ) %>%
  group_by(PriceCategory, Pattern) %>%
  summarise(moyenne = mean(AvgPlayers, na.rm = TRUE), .groups = 'drop')

# Création du graphique avec ggplot2 et ggpattern
p <- ggplot(jeux_categorises, aes(x = PriceCategory, y = moyenne,
                                  pattern = Pattern)) +
  geom_bar_pattern(stat = "identity", width = 0.7, pattern_density = 0.1,
                   pattern_spacing = 0.02) +
  scale_pattern_manual(
    values = unique(jeux_categorises$Pattern),
    labels = c("Note >= 70", "Note < 70"),
    guide = guide_legend(title = "Catégories de notes Métacritic\n")
  ) +
  theme_minimal() +
  labs(
    title = "Répartition du nombre moyen de joueurs
en fonction du prix et de la note Metacritic",
    x = "",
    y = "Nombre moyen de joueurs"
  ) +
  theme(
    legend.position = "right",
    legend.title.align = 0.5,
    legend.text = element_text(size = 9),
    legend.title = element_text(size = 10, face = "bold"),
    legend.key = element_blank()
  ) +
  coord_cartesian(ylim = c(0, 60000))

# Afficher le graphique
print(p)
```


Répartition du nombre moyen de joueurs
en fonction du prix et de la note Metacritic
Nombre moyen de joueurs
60000
40000
20000
0
Gratuit
Prix < 20€
Prix >= 20€
Catégories de notes Métacritic
Note >= 70
Note < 70