

Analyse des jeux vidéo 2024 sur Grouvee

Ibrahima Bodian

Contents

```
# Chargement des données
df<-read_delim("../fichiercsv/grouvee_2024_games.csv", delim = ";", locale = locale(encoding="Latin1"),

# Conversion de la colonne Note Moyenne en numérique
df$`Note Moyenne`<-as.numeric(df$`Note Moyenne`)

# Aperçu
glimpse(df)

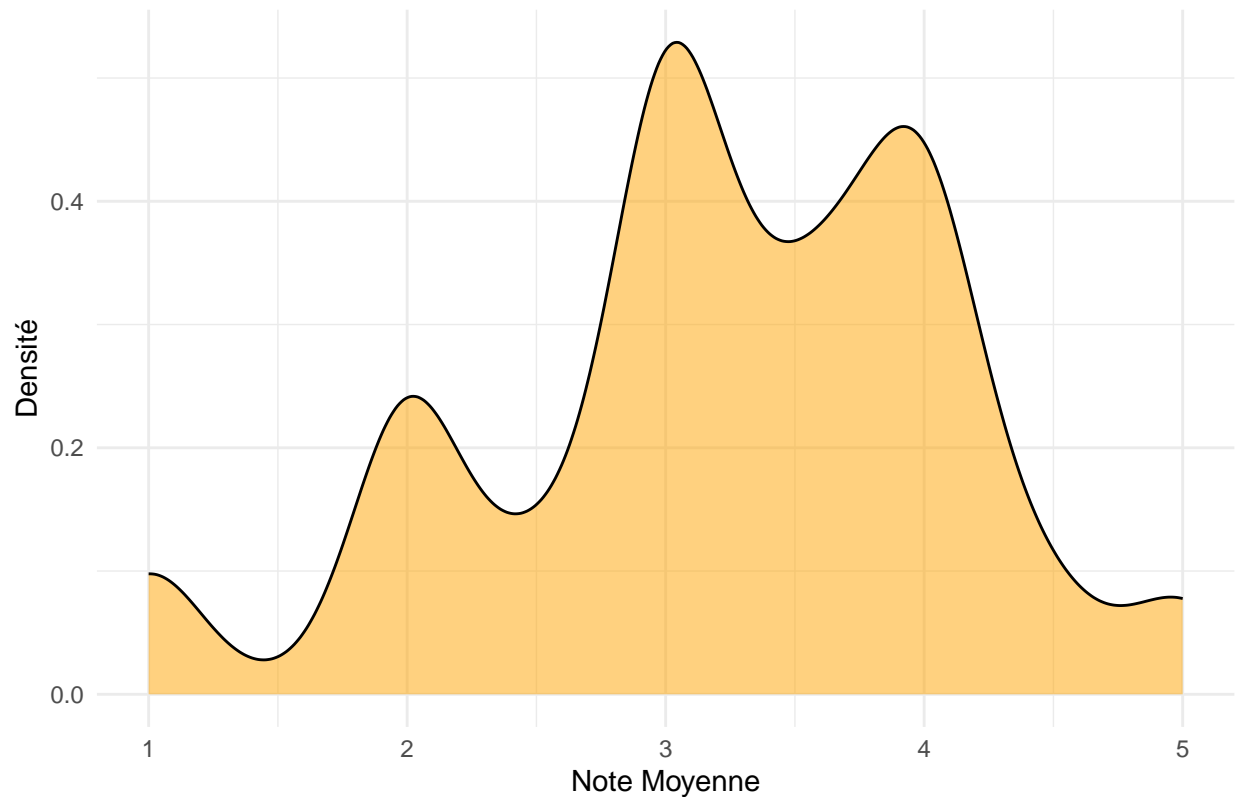
## Rows: 1,669
## Columns: 8
## $ Titre          <chr> "Balatro", "Elden Ring - Shadow of the Erdtree", "Fin~
## $ 'Date de sortie' <chr> "Feb 20, 2024", "Jan 01, 2024", "Feb 29, 2024", "Feb ~
## $ Développeur    <chr> "LocalThunk", NA, "Square Enix", "Atlus Co., Ltd. , P~
## $ Editeur        <chr> "PlayStack", NA, "Square Enix", "Sega", "Sony Compute~
## $ Genres          <chr> "Card Game", NA, "Action , Role-Playing", "Role-Playi~
## $ Franchise       <chr> NA, NA, "Compilation of Final Fantasy VII , Final Fan~
## $ 'Note Moyenne'  <dbl> 429, 45, 43, 44, 47, 401, 443, 399, 409, 413, 44, 334~
## $ Plateformes     <chr> "Nintendo Switch, PC, PlayStation 4, PlayStation 5, X~

# -----
# 1. Tendance générale des notes
# -----
# Chargement des données
df<-read_delim("../fichiercsv/grouvee_2024_games.csv", delim = ";", locale=locale(encoding = "Latin1"),

# Nettoyage : transformer les virgules en points et convertir en numérique
df$`Note Moyenne`<-gsub(",", ".", df$`Note Moyenne`)
df$`Note Moyenne`<-as.numeric(df$`Note Moyenne`)
df_notes<-df %>% filter(`Note Moyenne` > 0)

ggplot(df_notes, aes(x = `Note Moyenne`)) +
  geom_density(fill = "orange", alpha = 0.5, color = "black") +
  labs(title = "Courbe de densité des notes moyennes",
       x = "Note Moyenne", y = "Densité") +
  theme_minimal()
```

Courbe de densité des notes moyennes

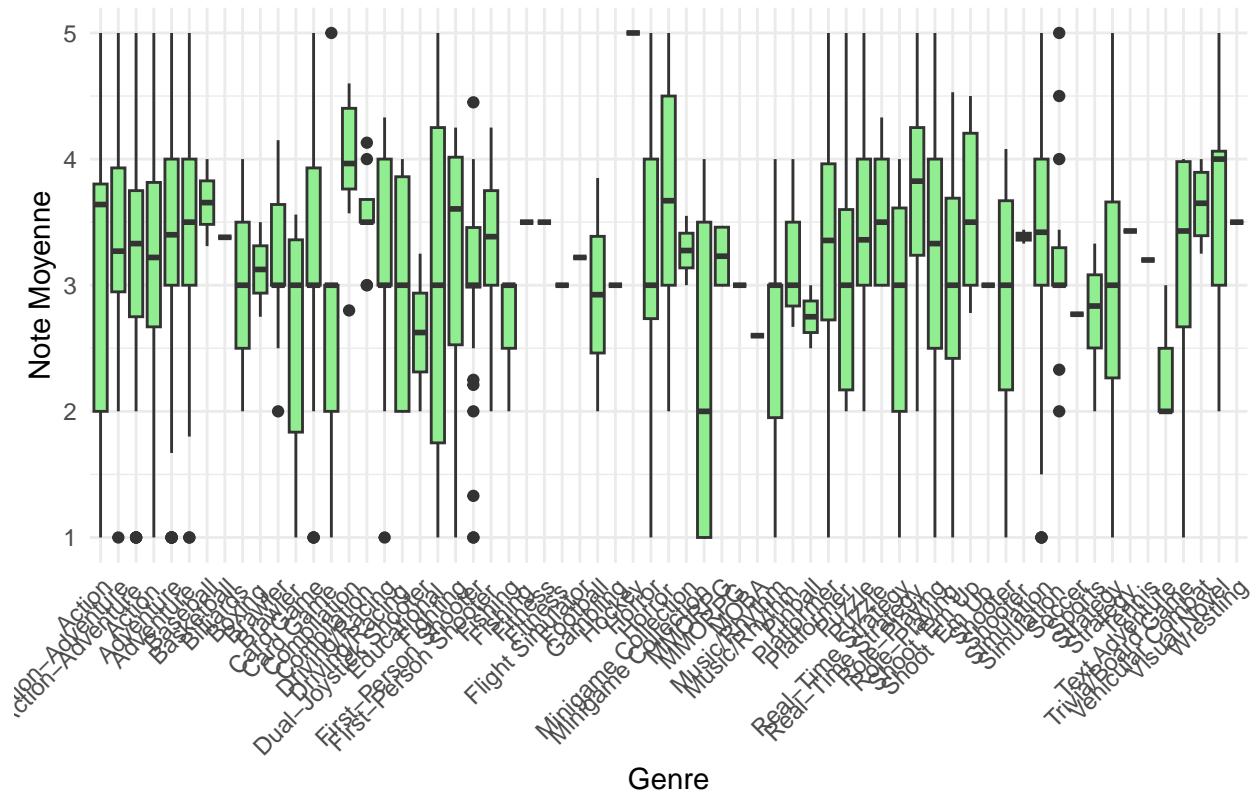


```
# -----
# 2. Comparaison des genres
# -----
df_genres<-df %>%
  filter(!is.na(Genres) & Genres != "") %>%
  separate_rows(Genres, sep = ",\\s*")

genre_summary<-df_genres %>%
  filter(`Note Moyenne` > 0) %>%
  group_by(Genres) %>%
  summarise(Note_Moyenne = mean(`Note Moyenne`, na.rm=TRUE), Nombre_jeux = n(), .groups = "drop") %>%
  arrange(desc(Note_Moyenne))

# Boxplot par genre
ggplot(df_genres %>% filter(`Note Moyenne` > 0), aes(x = Genres, y = `Note Moyenne`)) +
  geom_boxplot(fill = "lightgreen") +
  labs(title = "Répartition des notes par genre", x = "Genre", y = "Note Moyenne") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Répartition des notes par genre



Nettoyage et préparation des genres

```
df_genres<-df %>%
  filter(!is.na(Genres) & Genres != "") %>%
  separate_rows(Genres, sep = ",\\s*") %>%
  group_by(Genres) %>%
  summarise(Nombre_jeux = n(), .groups = "drop") %>%
  arrange(desc(Nombre_jeux))
```

Treemap sans légende

```
ggplot(df_genres, aes(area = Nombre_jeux, fill = Genres, label = paste0(Genres, "\n", Nombre_jeux))) +
  geom_treemap() +
  geom_treemap_text(colour = "white", place = "centre", grow = TRUE) +
  labs(title = "Répartition nombre de jeux par genre (Treemap)") +
  guides(fill = "none") + # Supprimer la légende
  theme_minimal()
```

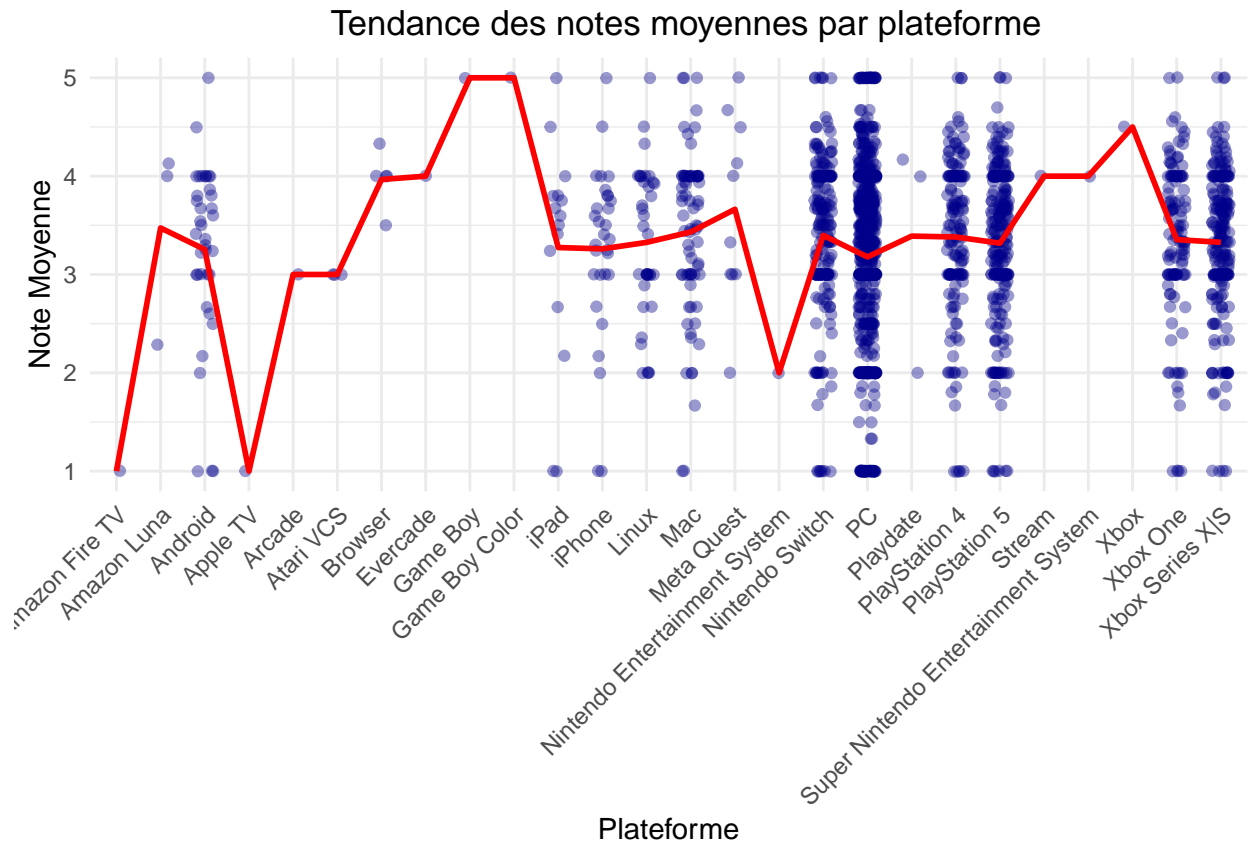
Répartition nombre de jeux par genre (Treemap)



```
# -----
# 3. Plateformes et notes
# -----
df_platforms<-df %>%
  filter(!is.na(Plateformes) & Plateformes != "") %>%
  separate_rows(Plateformes, sep = ",\\s*")

platform_summary<-df_platforms %>%
  filter(`Note Moyenne` > 0) %>%
  group_by(Plateformes) %>%
  summarise(Note_Moyenne=mean(`Note Moyenne`, na.rm=TRUE), Nombre_jeux = n(), .groups = "drop") %>%
  arrange(desc(Note_Moyenne))

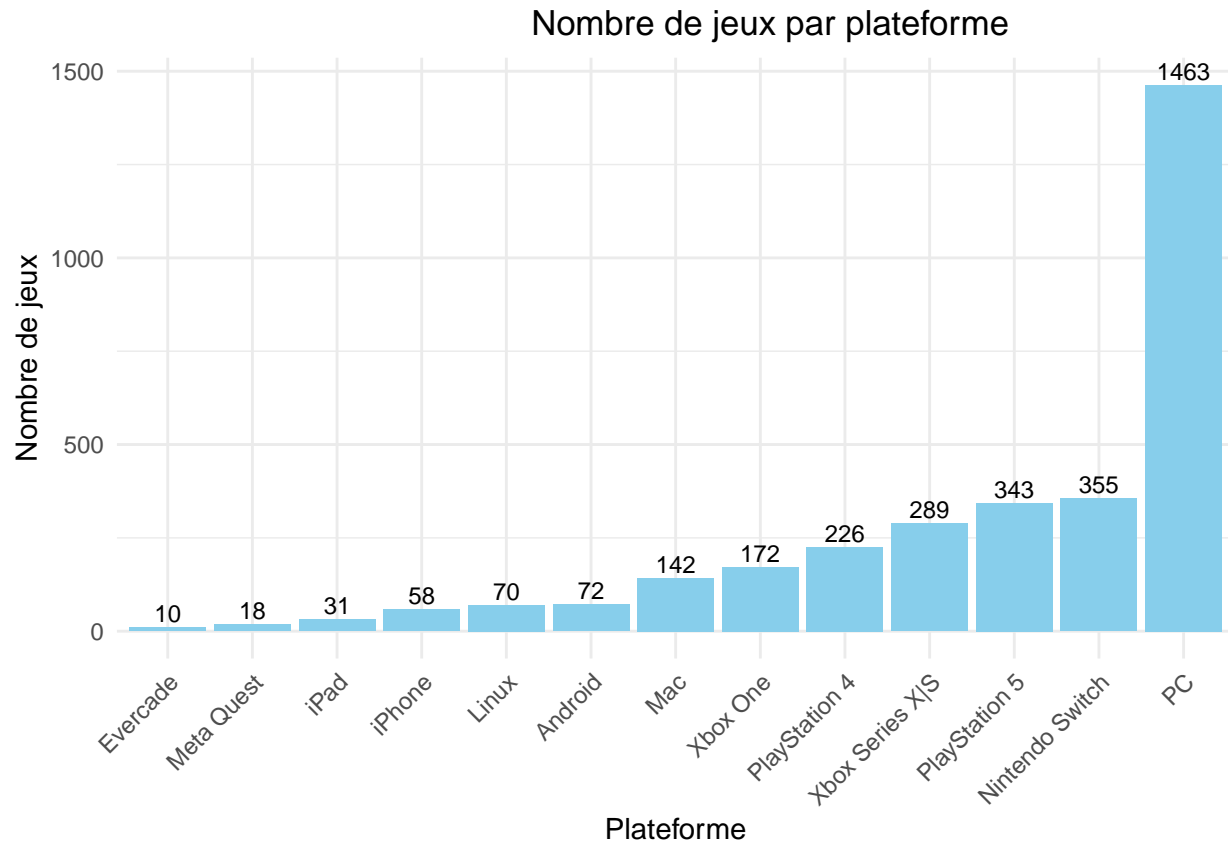
ggplot(df_platforms %>% filter(`Note Moyenne` > 0), aes(x = Plateformes, y = `Note Moyenne`)) +
  geom_jitter(width = 0.2, alpha = 0.4, color = "darkblue") +
  stat_summary(fun = mean, geom = "line", aes(group = 1), color = "red", linewidth = 1) +
  labs(title = "Tendance des notes moyennes par plateforme",
       x = "Plateforme", y = "Note Moyenne") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# -----
# Nombre de jeux par plateforme
# -----
# Séparation des plateformes multiples
df_platforms<-df %>%
  filter(!is.na(Plateformes) & Plateformes != "") %>%
  separate_rows(Plateformes, sep = ",\\s*")

# Calcul du nombre de jeux par plateforme
nb_jeux_par_plateforme<-df_platforms %>%
  group_by(Plateformes) %>%
  summarise(Nombre_jeux = n(), .groups = "drop") %>%
  filter(Nombre_jeux >= 10) %>% # <--- filtre ici
  arrange(Nombre_jeux)

# Graphique
ggplot(nb_jeux_par_plateforme, aes(x = reorder(Plateformes, Nombre_jeux), y = Nombre_jeux)) +
  geom_col(fill = "skyblue") +
  geom_text(aes(label = Nombre_jeux), vjust = -0.3, size = 3) +
  labs(title = "Nombre de jeux par plateforme",
       x = "Plateforme", y = "Nombre de jeux") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# -----
# 4. Clustering
# -----
# Préparer les données de clustering en gardant la colonne Titre pour les étiquettes
df_clust_all<-df %>%
  filter(`Note Moyenne` > 0 & !is.na(Plateformes)) %>%
  mutate(nb_plateformes = str_count(Plateformes, ",") + 1) %>%
  select(Titre, `Note Moyenne`, nb_plateformes)

# Mise à l'échelle des données numériques pour le clustering
df_clust_scaled<-scale(df_clust_all %>% select(`Note Moyenne`, nb_plateformes))

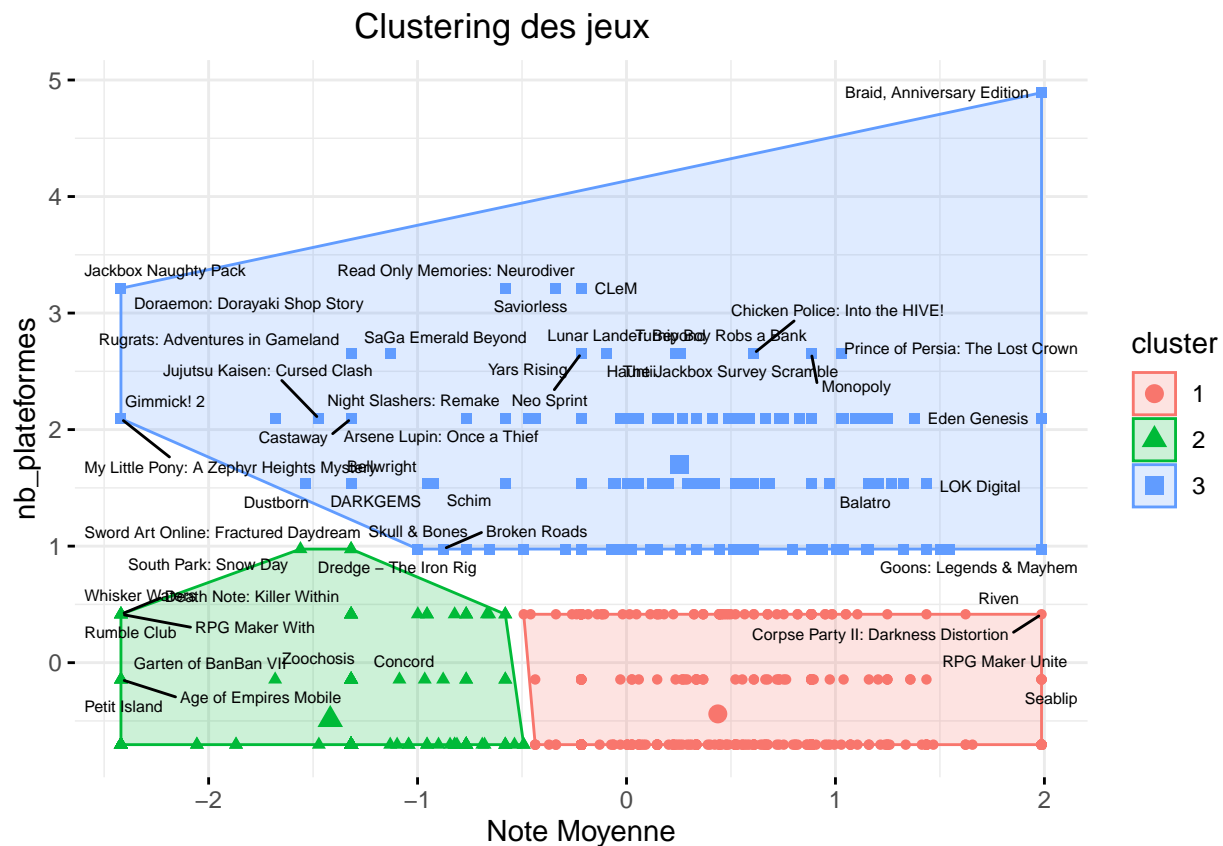
# Exécuter le clustering k-means avec 3 clusters
kmeans_result<-kmeans(df_clust_scaled, centers = 3, nstart = 25)

# Visualisation de base du clustering avec fviz_cluster
cluster_plot<-fviz_cluster(
  object = kmeans_result,
  data   = df_clust_scaled,
  geom   = "point",           # Tracer les points
  ellipse.type = "convex",    # Afficher l'enveloppe convexe des clusters
  label  = "none",           # Pas de labels automatiques
  ggtheme = theme_minimal(),
  main   = "Clustering des jeux"
)
```

```

# Ajouter les étiquettes manuellement avec geom_text_repel pour éviter le chevauchement
cluster_plot +
  geom_text_repel(
    aes(
      x = cluster_plot$data$x,
      y = cluster_plot$data$y,
      label = df_clust_all$Titre
    ),
    size = 2,          # Taille du texte
    max.overlaps = 20  # Limite le chevauchement des labels
  )

```



```

# -----
# 5. Corrélation
# -----
set.seed(123)

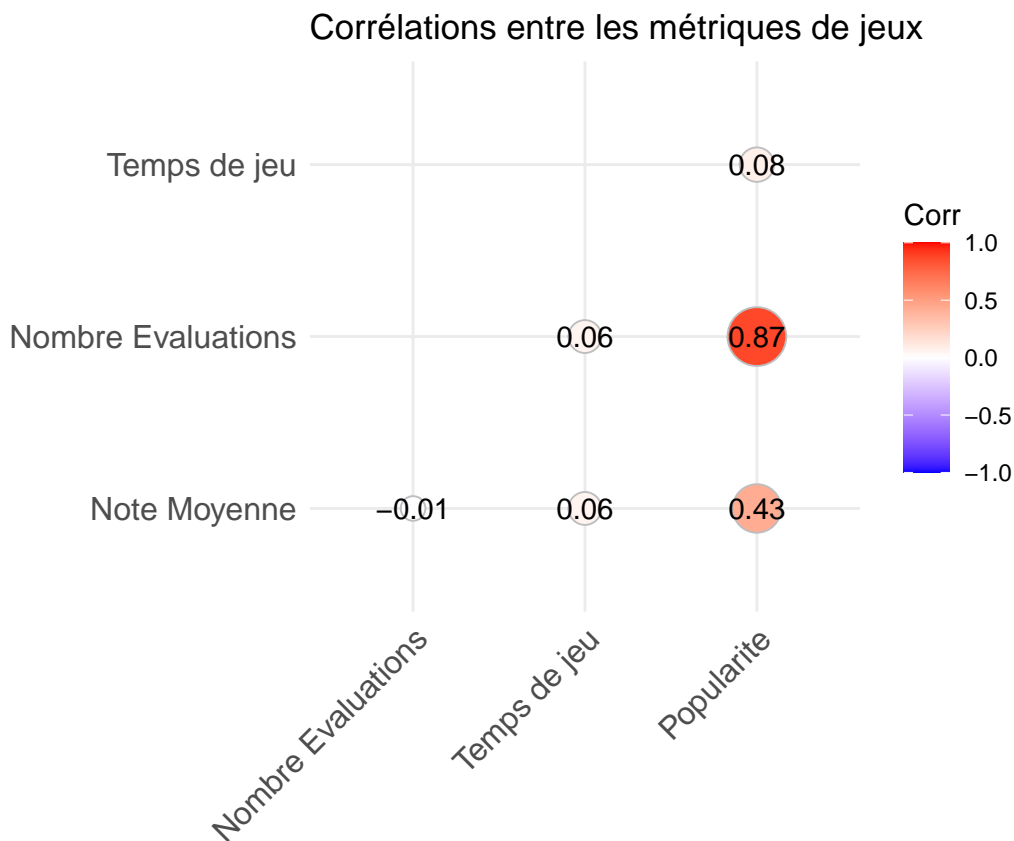
df_filtered<-df %>%
  filter(`Note Moyenne` > 0) %>%
  mutate(
    `Nombre Evaluations` = sample(10:1000, nrow(.), replace = TRUE),
    `Temps de jeu` = round(runif(nrow(.), 1, 100), 1),
    `Popularite` = round(`Note Moyenne` * `Nombre Evaluations`)
  )

```

```
df_corr<-df_filtered %>%
  select(`Note Moyenne`, `Nombre Evaluations`, `Temps de jeu`, Popularite) %>%
  mutate(across(everything(), as.numeric)) %>%
  na.omit()

cor_matrix<-cor(df_corr)

ggcorrplot(cor_matrix,
  method = "circle",
  type = "lower",
  lab = TRUE,
  colors = c("blue", "white", "red"),
  title = "Corrélations entre les métriques de jeux",
  ggtheme = theme_minimal())
```



```
# -----
# 6. Franchises
# -----

# Importer les données avec la colonne "Note Moyenne" en texte
df<-read_delim("../fichiercsv/grouvee_2024_games.csv", delim = ";", locale = locale(encoding = "Latin1"))

# Correction : Remplacer les virgules par des points et convertir en numérique
df$`Note Moyenne`<-as.numeric(gsub(",", ".", df$`Note Moyenne`))
```

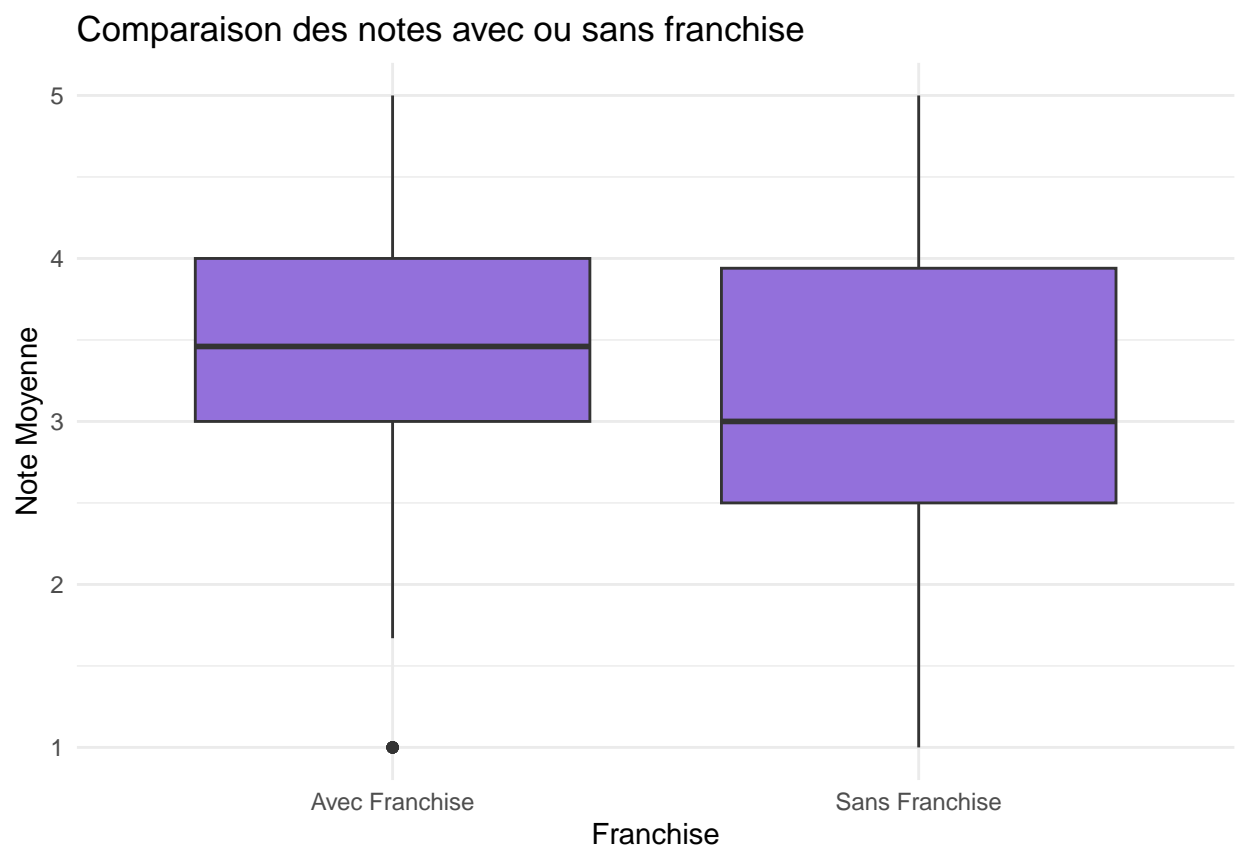


```

# Créer une nouvelle variable "Avec_Franchise" :
# Si la colonne Franchise est vide ou NA, on considère que le jeu est "Sans Franchise", sinon "Avec Franchise"
df_franchise<-df %>%
  mutate(Avec_Franchise = if_else(is.na(Franchise) | Franchise == "", "Sans Franchise", "Avec Franchise"))
  filter(`Note Moyenne` > 0) # Ne conserver que les jeux notés (> 0)

# Boxplot comparatif des notes selon l'appartenance à une franchise
ggplot(df_franchise, aes(x = Avec_Franchise, y = `Note Moyenne`)) +
  geom_boxplot(fill = "mediumpurple") +
  labs(title = "Comparaison des notes avec ou sans franchise",
       x = "Franchise", y = "Note Moyenne") +
  theme_minimal()

```



```

# -----
# 7. Régression linéaire
# -----

df_reg<-df %>%
  filter(`Note Moyenne` > 0 & !is.na(Plateformes)) %>%
  mutate(nb_plateformes = str_count(Plateformes, ",") + 1)

# Graphique
ggplot(df_reg, aes(x = nb_plateformes, y = `Note Moyenne`)) +
  geom_point(color = "blue", alpha = 0.6) +

```

```
geom_smooth(method = "lm", se = FALSE, color = "red") +
labs(title = "Régression : Note Moyenne vs Nombre de plateformes") +
theme_minimal()
```

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

