

## Analyse des jeux vidéo 2024 sur Grouvee

```
# 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~

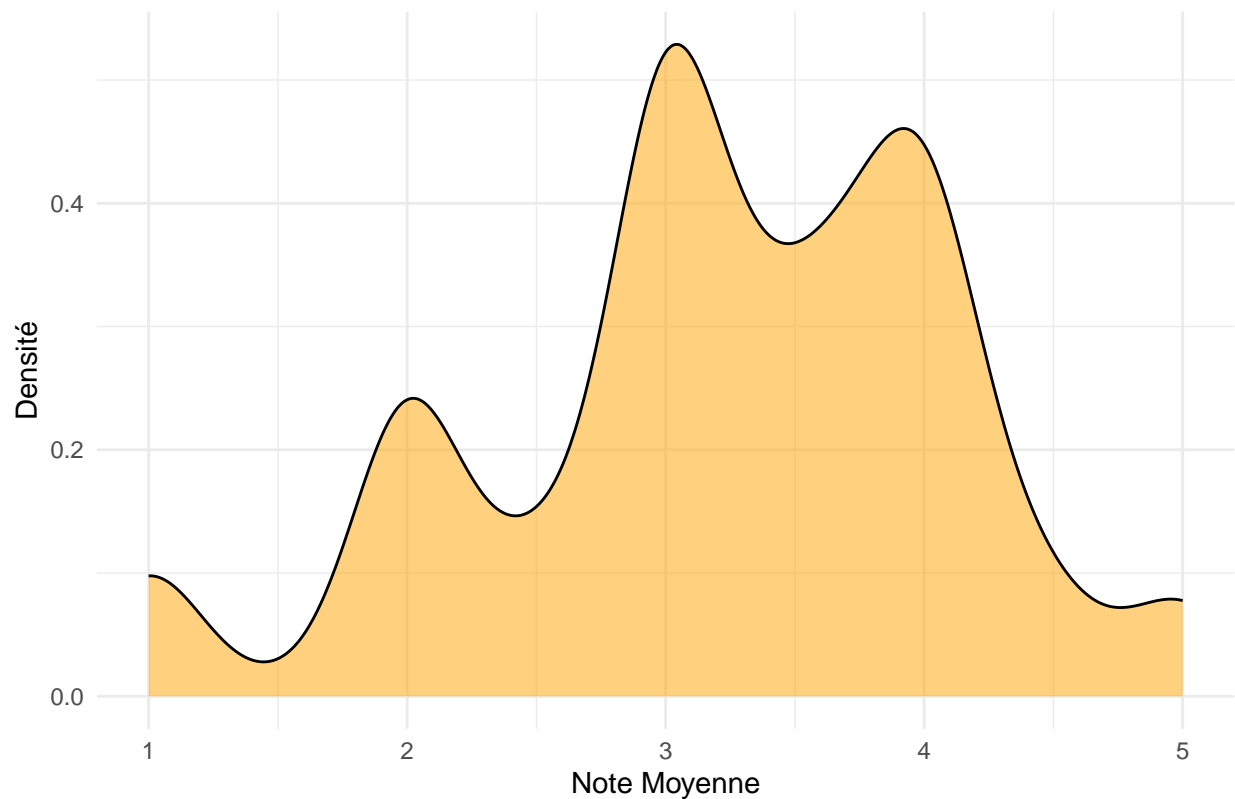
# -----
# 1. Tendance générale des notes
# -----
# Chargement des données
df <- read_delim("../fichiercsv/grouvee_2024_games.csv",
                  delim = ";",
                  locale = locale(encoding = "Latin1"),
                  col_types = cols(`Note Moyenne` = col_character()))

# Nettoyage : transformer les virgules en points et convertir en numérique
df$`Note Moyenne` <- gsub(",", ".", df$`Note Moyenne`) # Remplacer la virgule par un point
df$`Note Moyenne` <- as.numeric(df$`Note Moyenne`)     # Conversion en numérique

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



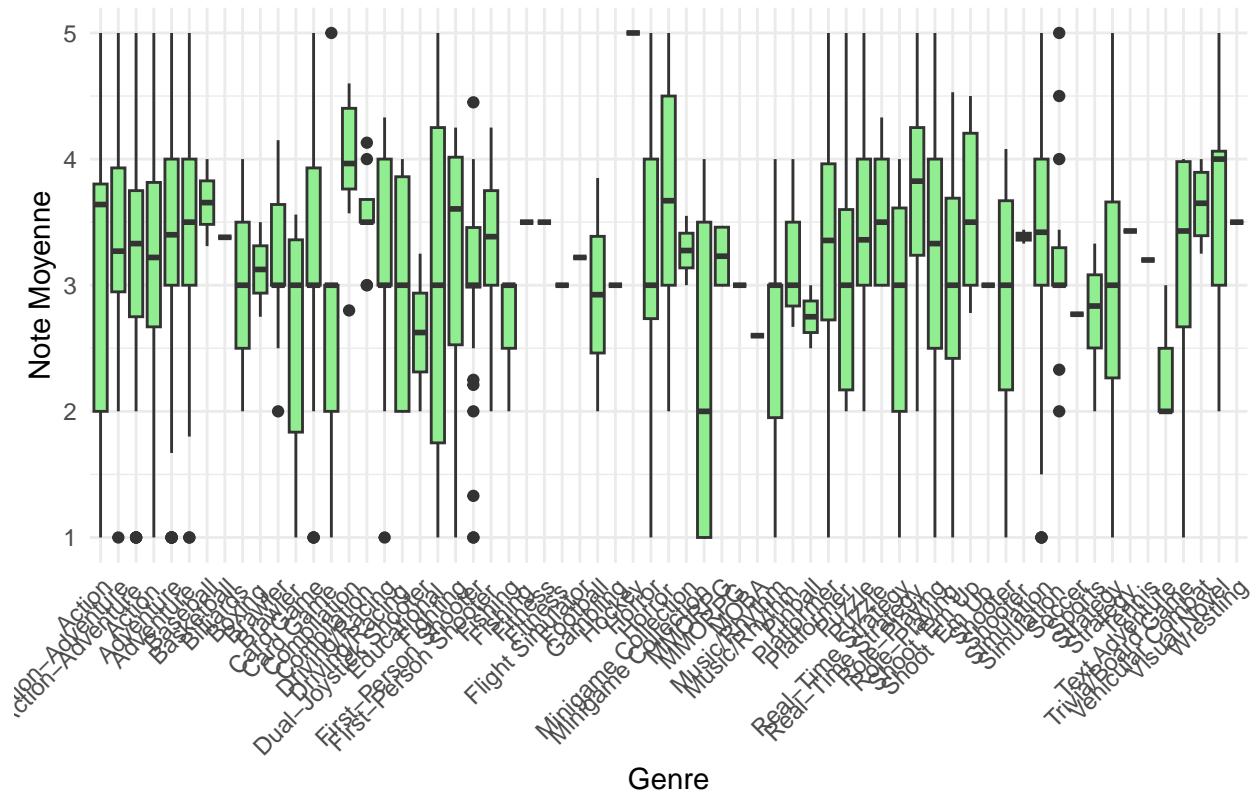
```
# -----
# 2. Comparaison des genres
# -----

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)

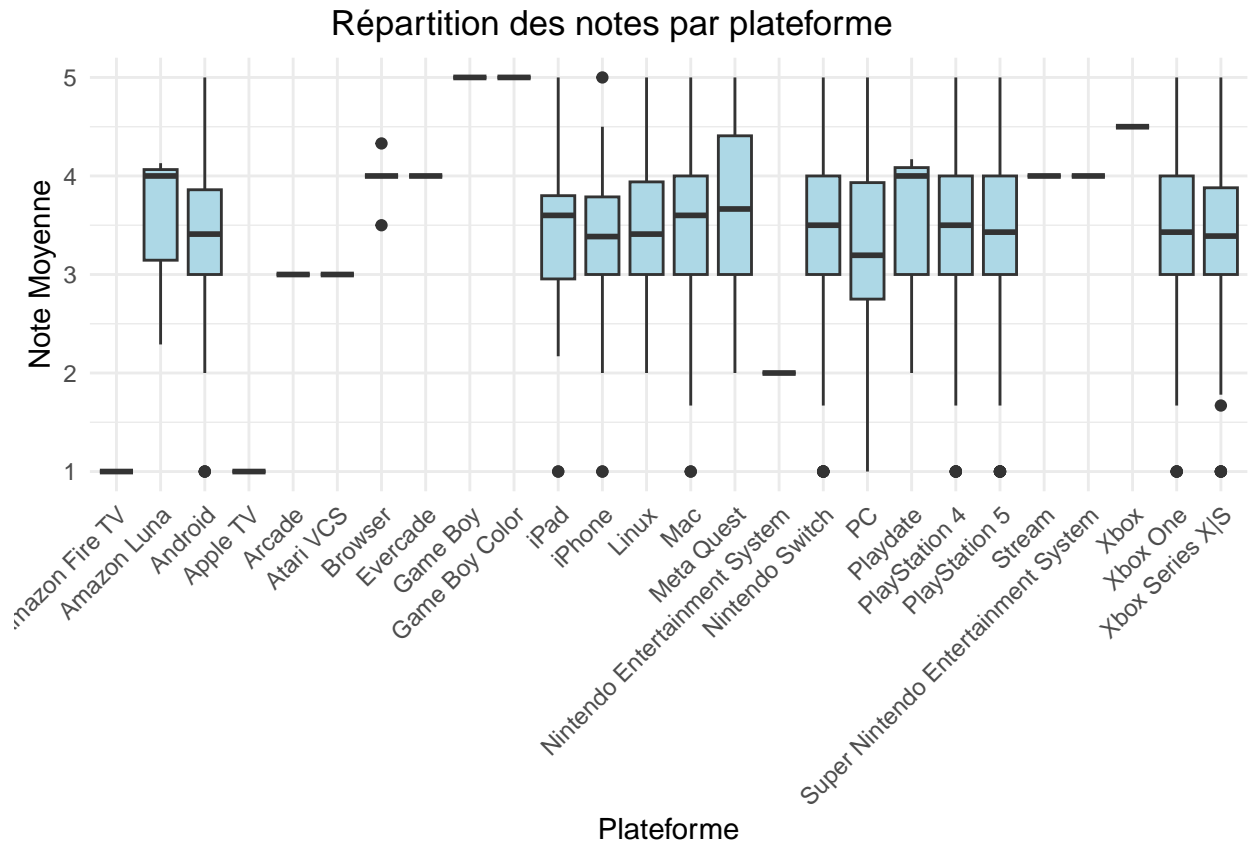


```
# -----
# 3. Plateformes et notes
# -----

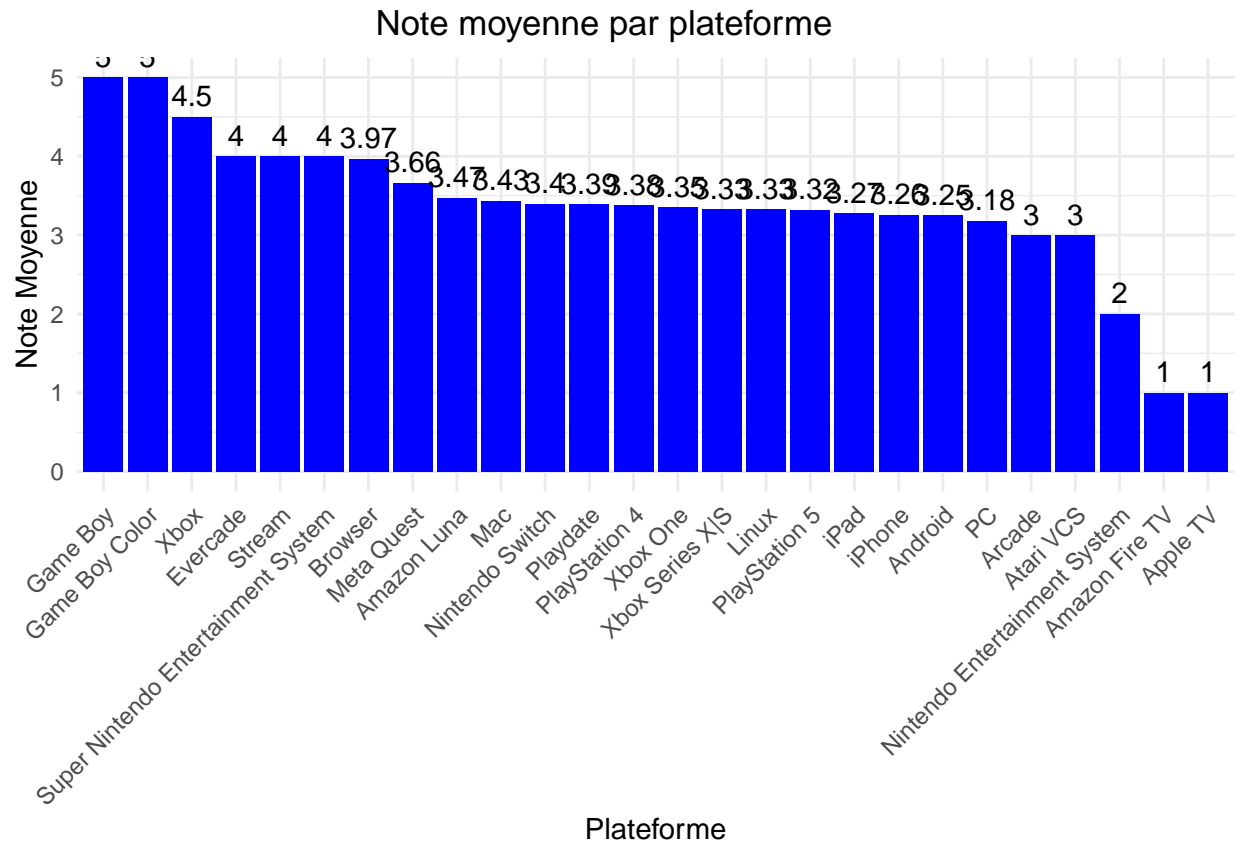
df_platforms<-df %>%
  filter(!is.na(Plateformes) & Plateformes != "") %>%
  separate_rows(Plateformes, sep = ",\\s*")

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))

# Boxplot par plateforme
ggplot(df_platforms %>% filter(`Note Moyenne` > 0), aes(x = Plateformes, y = `Note Moyenne`)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Répartition des notes par plateforme", x = "Plateforme", y = "Note Moyenne") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# Moyenne par plateforme
ggplot(platform_summary, aes(x = reorder(Plateformes, -Note_Moyenne), y = Note_Moyenne)) +
  geom_col(fill = "blue") +
  geom_text(aes(label = round(Note_Moyenne, 2)), vjust = -0.5) +
  labs(title = "Note moyenne par plateforme", x = "Plateforme", y = "Note Moyenne") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



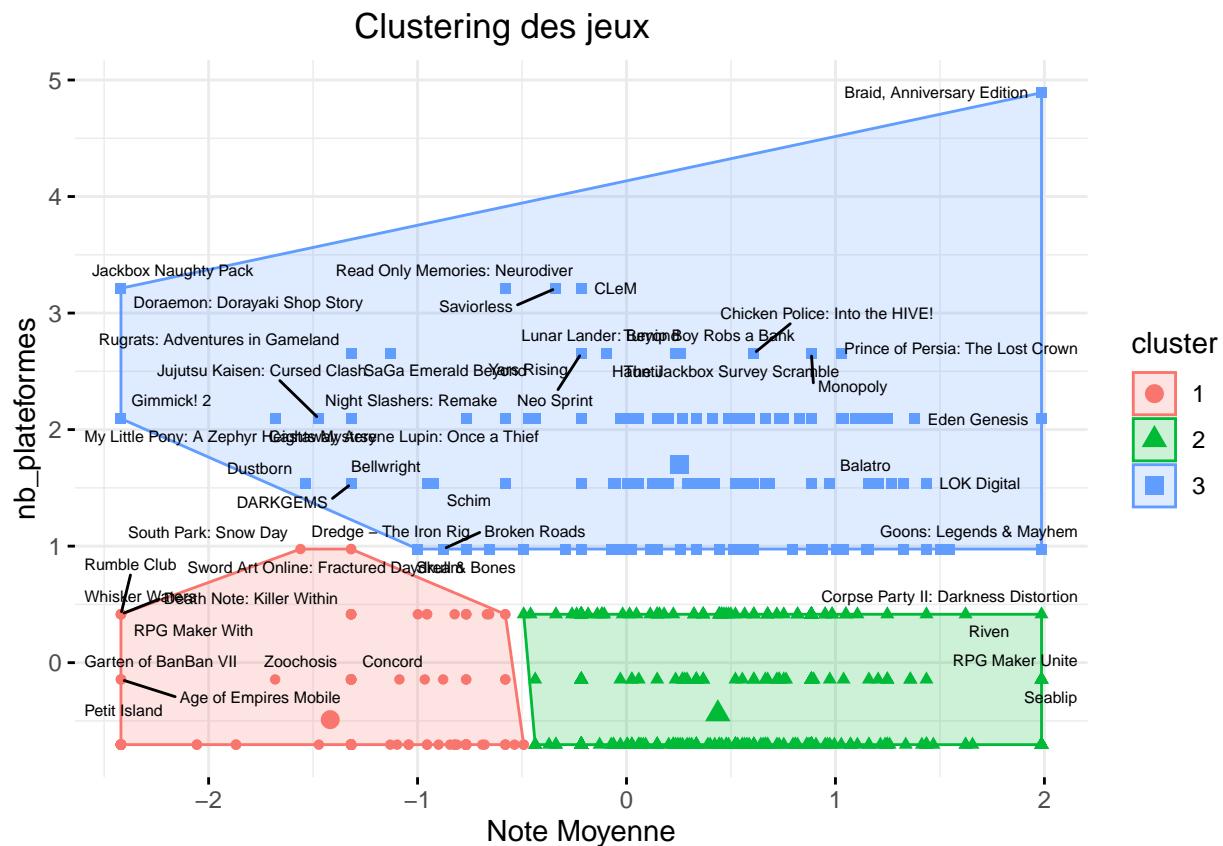
```
# -----
# 4. Clustering
# -----
# Préparer les données de clustering en gardant la colonne Titre pour les étiquettes
df_clust_all <- df %>%
  filter(`Note Moyenne` > 0 & !is.na(Plateformes)) %>%
  mutate(nb_plateformes = str_count(Plateformes, ",") + 1) %>%
  select(Titre, `Note Moyenne`, nb_plateformes)

# 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
# -----

# Filtrer les jeux notés (Note Moyenne > 0) et extraire nb_plateformes
df_corr <- df %>%
  filter(`Note Moyenne` > 0 & !is.na(Plateformes)) %>%
  mutate(nb_plateformes = str_count(Plateformes, ",") + 1) %>%
  select(`Note Moyenne`, nb_plateformes)

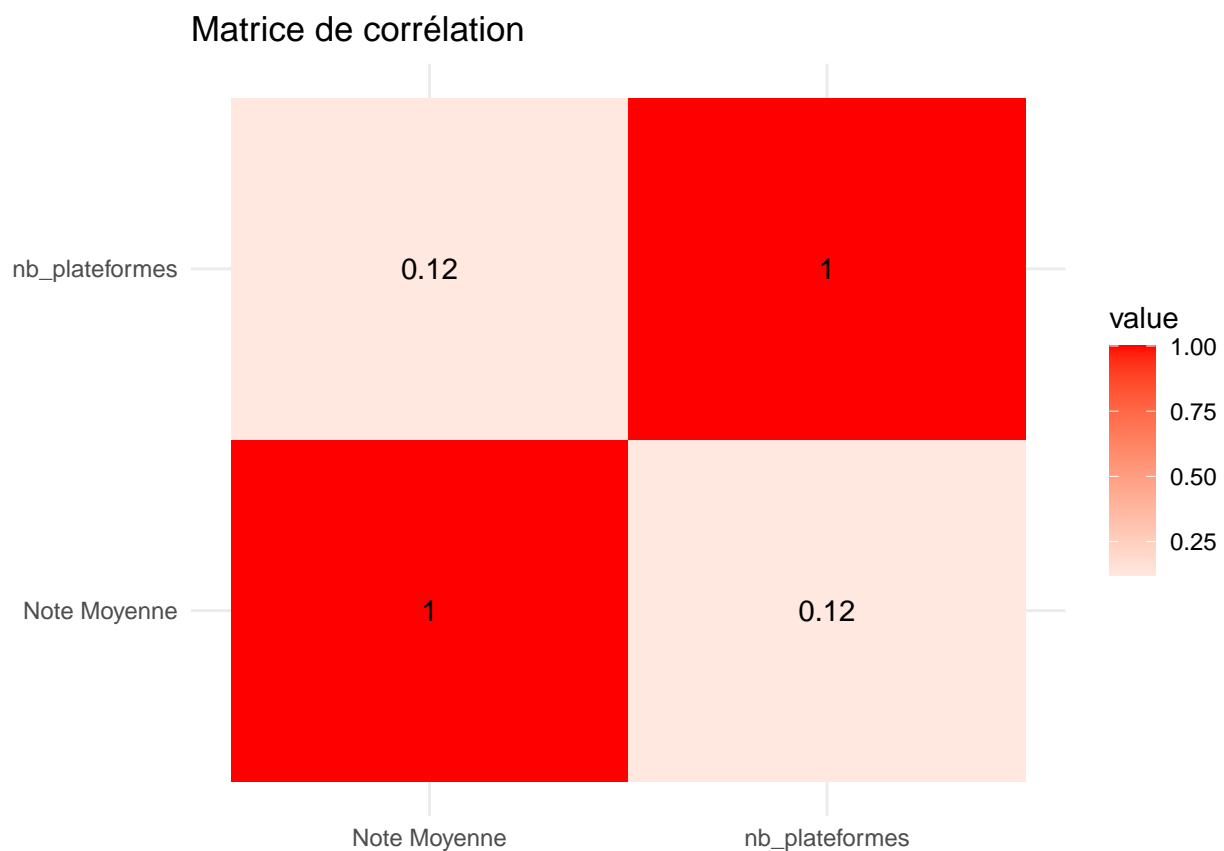
# Calcul de la matrice de corrélation (seulement 2 variables ici)
cor_matrix <- cor(df_corr, use = "complete.obs")
```

```

# Transformation de la matrice de corrélation au format long
melted_corr <- melt(cor_matrix)

# Création de la heatmap
ggplot(melted_corr, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  geom_text(aes(label = round(value, 2)), color = "black") + # Pour afficher la valeur numérique
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  labs(
    title = "Matrice de corrélation",
    x = NULL,
    y = NULL
  ) +
  theme_minimal()

```



```

# Chargement des données en forçant Note Moyenne comme texte
df<-read_delim("../fichiercsv/grouvee_2024_games.csv",
  delim = ";",
  locale = locale(encoding = "Latin1"),
  col_types = cols(`Note Moyenne` = col_character()))

# 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_sep <- df %>%

```

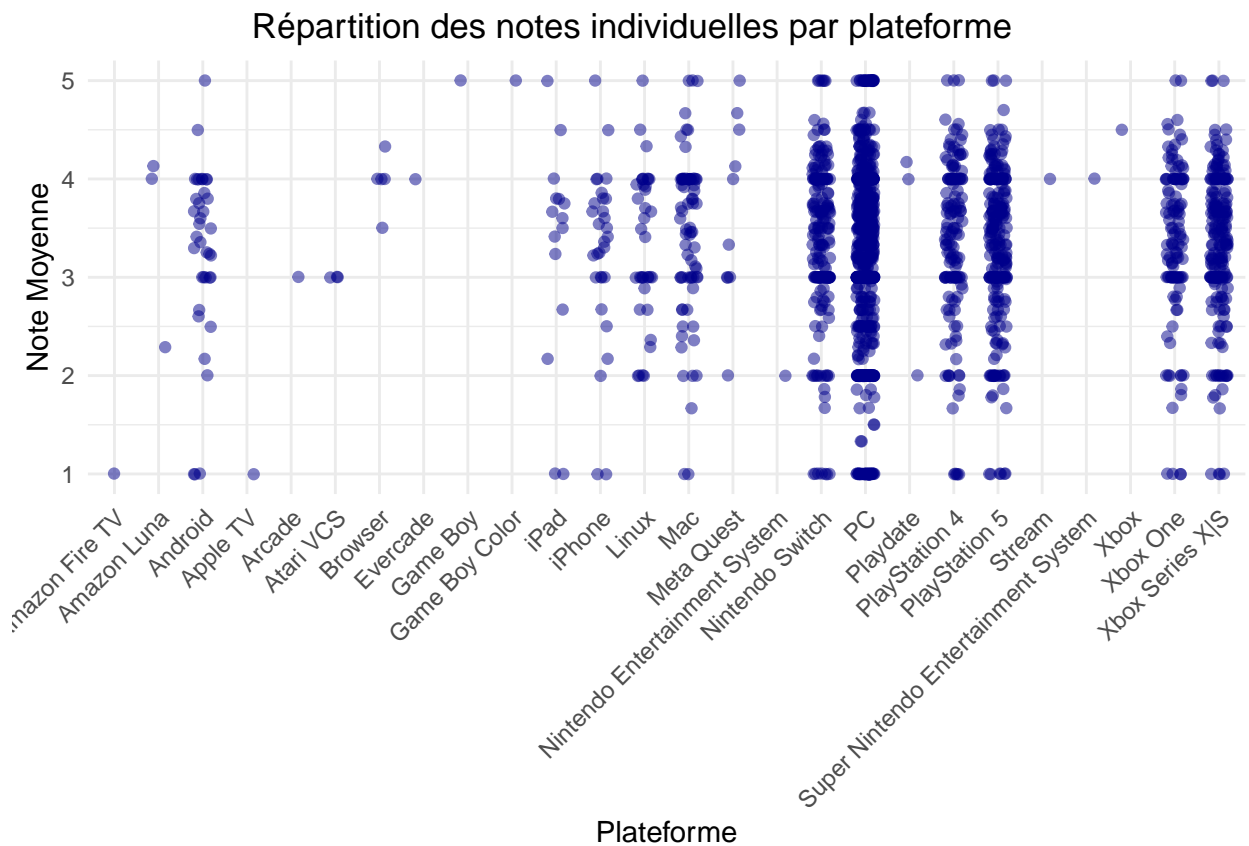


```

filter(`Note Moyenne` > 0 & !is.na(Plateformes)) %>%
separate_rows(Plateformes, sep = ",\\s*")

# 3. Afficher le jitter (notes entre 0 et 5 normalement)
ggplot(df_notes_sep, aes(x = Plateformes, y = `Note Moyenne`)) +
  geom_jitter(width = 0.2, alpha = 0.5, color = "darkblue") +
  labs(title = "Répartition des notes individuelles par plateforme",
       x = "Plateforme", y = "Note Moyenne") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```



```

# -----
# 8. 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"))

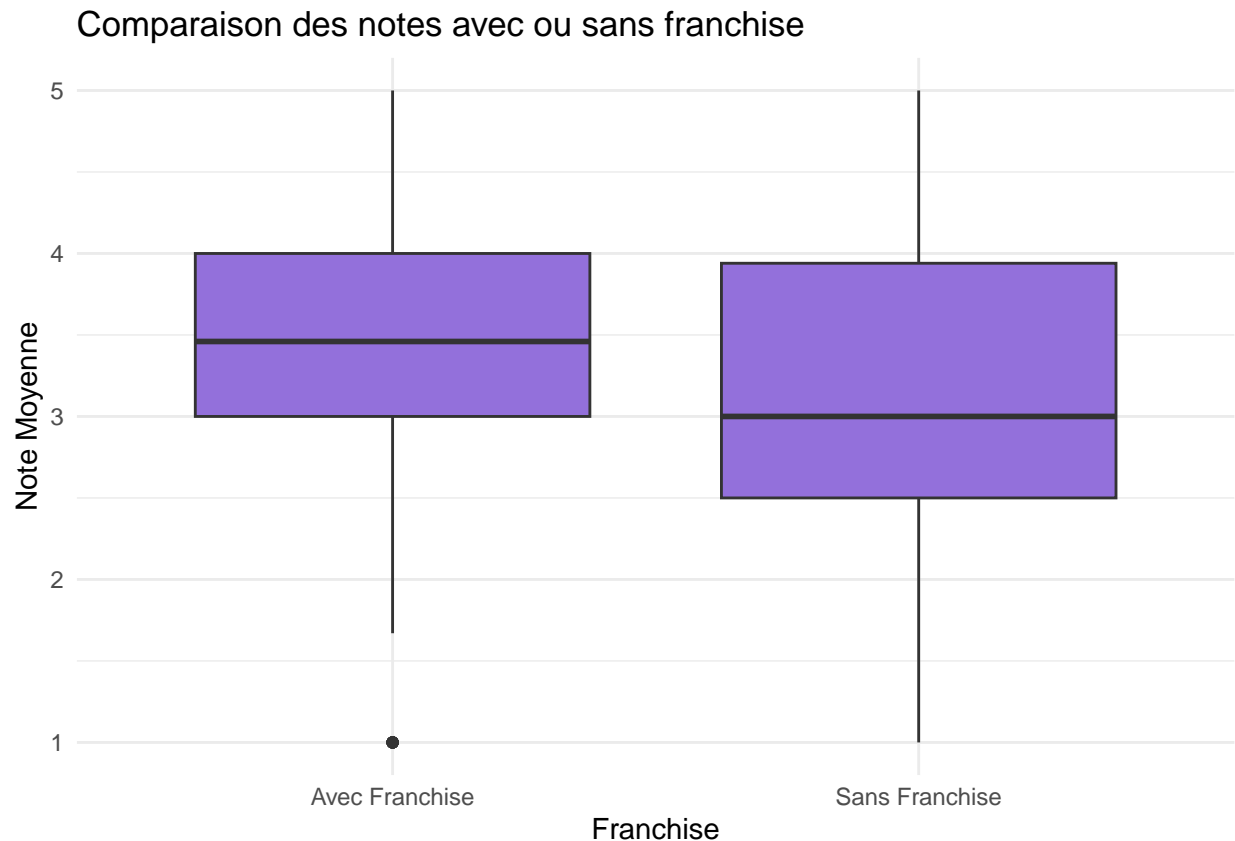
```

```

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()

```



```

# -----
# 9. 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'
```

