

Carnet analyse statistique LUIS, RENOUX, BOUREMANI

June 19, 2025

```
[1]: # A décommenter à la première exécution si jupyter se plaint de ne pas trouver
      ↳ oracledb
      #!pip install --upgrade oracledb
```

```
[10]: # Compléter ici les imports dont vous avez besoin, ne pas modifier ceux déjà
      ↳ présents
import getpass
from os import getenv
import pandas as pd
import oracledb
import warnings
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2]: # Nécessaire pour éviter les problèmes de session
class Connexion(object):
    def __init__(self, login, password):
        self.conn = oracledb.connect(
            user=login,
            password=password,
            host="oracle.iut-orsay.fr",
            port=1521,
            sid="etudom",
        )
        self.conn.autocommit = False

    def __enter__(self):
        self.conn.autocommit = False
        return self.conn

    def __exit__(self, *args):
        self.conn.close()
```

```
[3]: # La fonction ci-dessous est à utiliser pour exécuter une requête et stocker
      ↳ les résultats dans un dataframe Pandas sans afficher d'alerte.
      # Vous pouvez vous en inspirer pour créer vos propres fonctions.
def requete_vers_dataframe(connexion_data, requete, valeurs = None):
```

```

with Connexion(login=connexion_data['login'],
↳password=connexion_data['password']) as connexion:
    warnings.simplefilter(action='ignore', category=UserWarning)
    if valeurs is not None:
        df = pd.read_sql(requete, connexion, params=valeurs)
    else:
        df = pd.read_sql(requete, connexion)
    warnings.simplefilter("always")
    return df

```

```

[4]: # Saisir ci-dessous la plateforme qui vous a été attribuée. Cela correspond au
↳nomPlateforme dans la table PLATEFORME de la base de données
# Par exemple NOM_PLATEFORME = "Switch"
NOM_PLATEFORME = "Sega Saturn"
# Saisir ci-dessous le login court de la base utilisée pour votre carnet
SCHEMA = "ILUIS"

```

```

“python # Gestion des droits de ANALYSEJV déjà effectué lorsque connecté
par iluis tables = [ “CATEGORIEJEU”, “CATEGORIEPLATEFORME”,
“CLASSIFICATIONAGE”, “COMPAGNIE”, “DATESORTIE”, “FRANCHISE”,
“JEU”, “GENRE”, “MODALITE”, “MODEMULTIJOUEUR”, “MOTCLE”,
“MOTEUR”, “PLATEFORME”, “POPULARITE”, “REGION”, “THEME”,
“TITREALTERNATIF”, “LOCALISATIONJEU”, “MOTCLEJEU”, “GENREJEU”,
“SIMILARITE”, “CLASSIFICATIONJEU”, “COMPAGNIEJEU”, “MOTEURJEU”,
“PLATEFORMEMOTEUR”, “THEMEJEU”, “FRANCHISEJEU”, “MODALITEJEU”]

```

```

with Connexion(login=conn['login'], password=conn['password']) as connexion: connexion.begin()
curseur = connexion.cursor() utilisateur = “ANALYSEJV”

```

```

for table in tables:
    sql_grant = f"""
    GRANT SELECT ON {SCHEMA}.{table} TO {utilisateur}
    """
    curseur.execute(sql_grant)

```

```

sql_grant = f"""
GRANT EXECUTE ON {SCHEMA}.DETAILS_SORTIES TO {utilisateur}
"""
curseur.execute(sql_grant)
connexion.commit()
curseur.close()

```

```

print(“ Droits accordés à l'utilisateur AnalyseJV”)

```

```

[170]: # On demande à l'utilisateur son login et mot de passe pour pouvoir accéder à
↳la base de données
if getenv("DB_LOGIN") is None:
    login = input("Login : ")
else:

```

```

    login = getenv("DB_LOGIN")
if getenv("DB_PASS") is None:
    password = getpass.getpass("Mot de passe : ")
else:
    password = getenv("DB_PASS")
conn = {'login': login, 'password': password}

```

```

Login : iluis
Mot de passe : .....

```

```

[171]: # On vérifie que l'utilisateur est bien connecté à la base de données, que le
      ↪ schéma est bon, et qu'on trouve la bonne édition des JO
data = requete_vers_dataframe(conn, f"SELECT * FROM {SCHEMA}.PLATEFORME WHERE
      ↪ nomPlateforme LIKE (:libelle)", {"libelle": NOM_PLATEFORME})
id_plateforme = int(data.IDPLATEFORME.iloc[0])
print(f"Identifiant de la plateforme : {id_plateforme}")

```

```

Identifiant de la plateforme : 32

```

1 TABLEAU DE analyse statistique

2 1. Présentation générale de la plateforme : Sega Saturn

2.1 Contexte historique

La **Sega Saturn** est une console de jeux vidéo de cinquième génération développée par la compagnie japonaise **SEGA**. Elle est sortie au Japon en **novembre 1994**, en Amérique du Nord en **mai 1995** et en Europe en **juillet 1995**.

La Saturn succède à la **Sega Mega Drive (Genesis)** et précède la **Sega Dreamcast**, dernière console de Sega. Elle fait donc partie de la **deuxième génération de consoles SEGA**, après les Master System et Mega Drive.

2.1.1 Caractéristiques principales :

- Support : **CD-ROM**
- Architecture : **bi-processeur** (ce qui a complexifié le développement)
- Concurrents directs : **Sony PlayStation, Nintendo 64**
- Commercialisation active : **1994 – 2000**

Malgré un succès notable au Japon, la Saturn n'a pas connu le même engouement international que ses concurrentes, notamment à cause d'un marketing confus et de difficultés techniques pour les développeurs tiers.

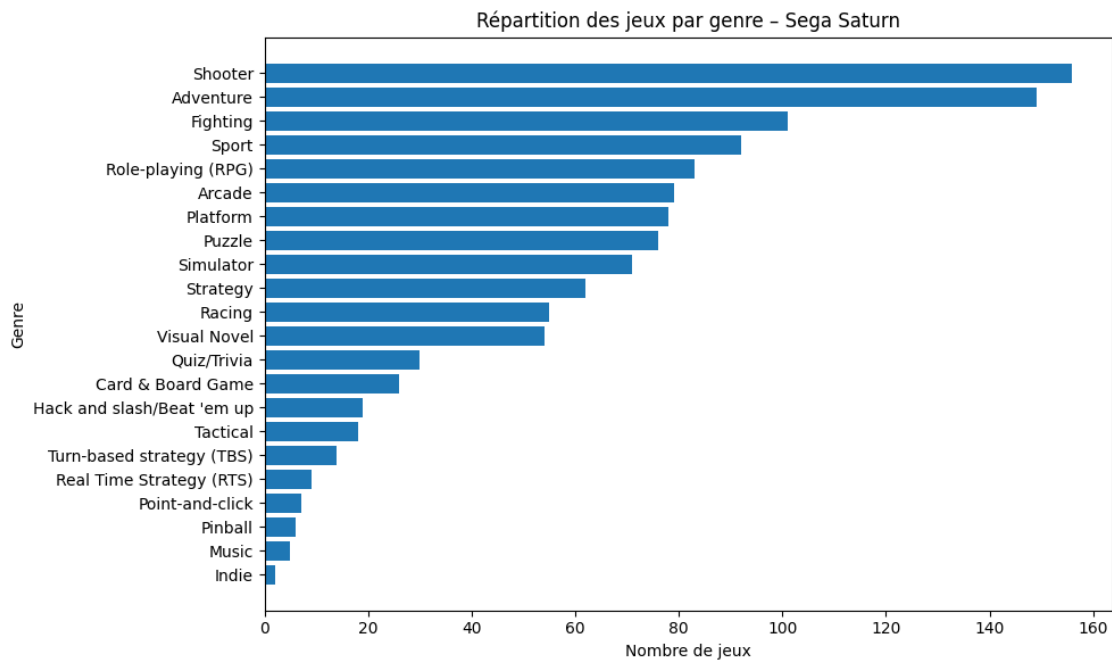
- Wikipédia – [Sega Saturn](#)
- IGN – Console History: <https://www.ign.com>
- RetroGaming Wiki – Sega Saturn Overview

2.2 Statistiques générales des jeux sur la Sega Saturn

3 1. Répartition des jeux par genre

```
[172]: data_genre = requete_vers_dataframe(
    conn,
    f"""
    SELECT g.nomGenre, COUNT(*) AS nbJeuX
    FROM {SCHEMA}.JEU j
    JOIN {SCHEMA}.GENREJEU gj ON j.idJeu = gj.idJeu
    JOIN {SCHEMA}.GENRE g ON gj.idGenre = g.idGenre
    JOIN {SCHEMA}.DATESORTIE ds ON ds.idJeu = j.idJeu
    WHERE ds.idPlateforme = 32
    GROUP BY g.nomGenre
    ORDER BY nbJeuX DESC
    """
)

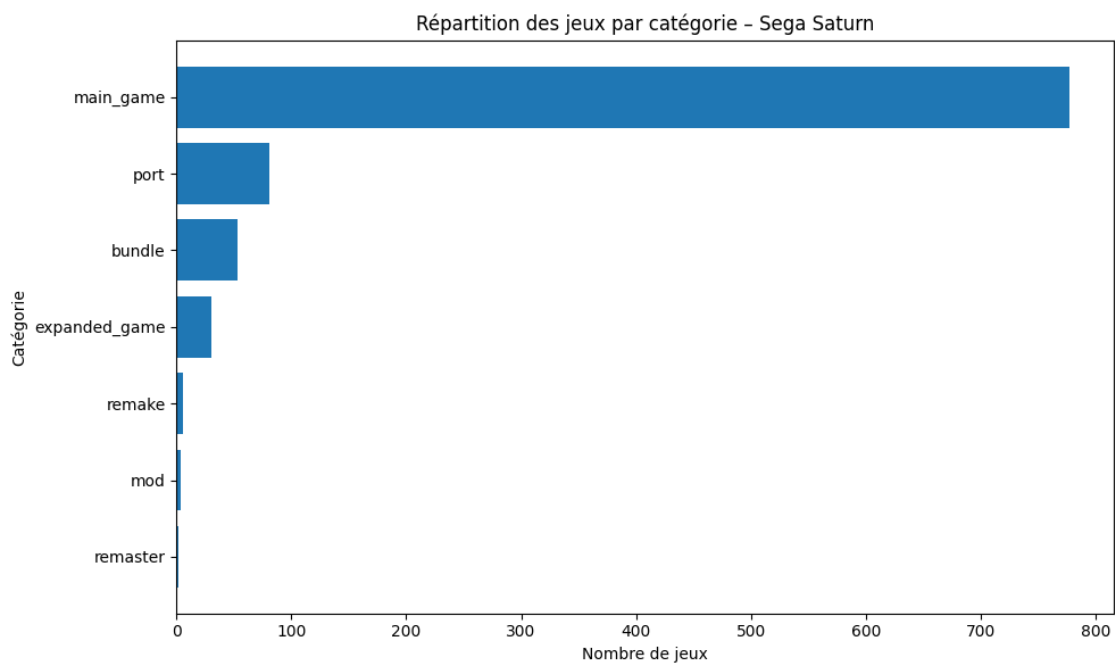
plt.figure(figsize=(10, 6))
plt.barh(data_genre["NOMGENRE"], data_genre["NBJEUX"])
plt.xlabel("Nombre de jeux")
plt.ylabel("Genre")
plt.title("Répartition des jeux par genre - Sega Saturn")
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```



Le genre du Shooter n'était pas très connue, mais vers les années 90, le genre c'est popularisé et c'est pour ça que le genre le plus produit sur Sega Neptun et le genre du shooter

```
[173]: data_cat = requete_vers_dataframe(
    conn,
    f"""
    SELECT cj.nomCategorieJeu, COUNT(*) AS nbJeux
    FROM {SCHEMA}.JEU j
    JOIN {SCHEMA}.CATEGORIEJEU cj ON j.CategorieJeu = cj.idCategorieJeu
    JOIN {SCHEMA}.DATESORTIE ds ON ds.idJeu = j.idJeu
    WHERE ds.idPlateforme = 32
    GROUP BY cj.nomCategorieJeu
    ORDER BY nbJeux DESC
    """
)

plt.figure(figsize=(10, 6))
plt.barh(data_cat["NOMCATEGORIEJEU"], data_cat["NBJEUX"])
plt.xlabel("Nombre de jeux")
plt.ylabel("Catégorie")
plt.title("Répartition des jeux par catégorie - Sega Saturn")
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```

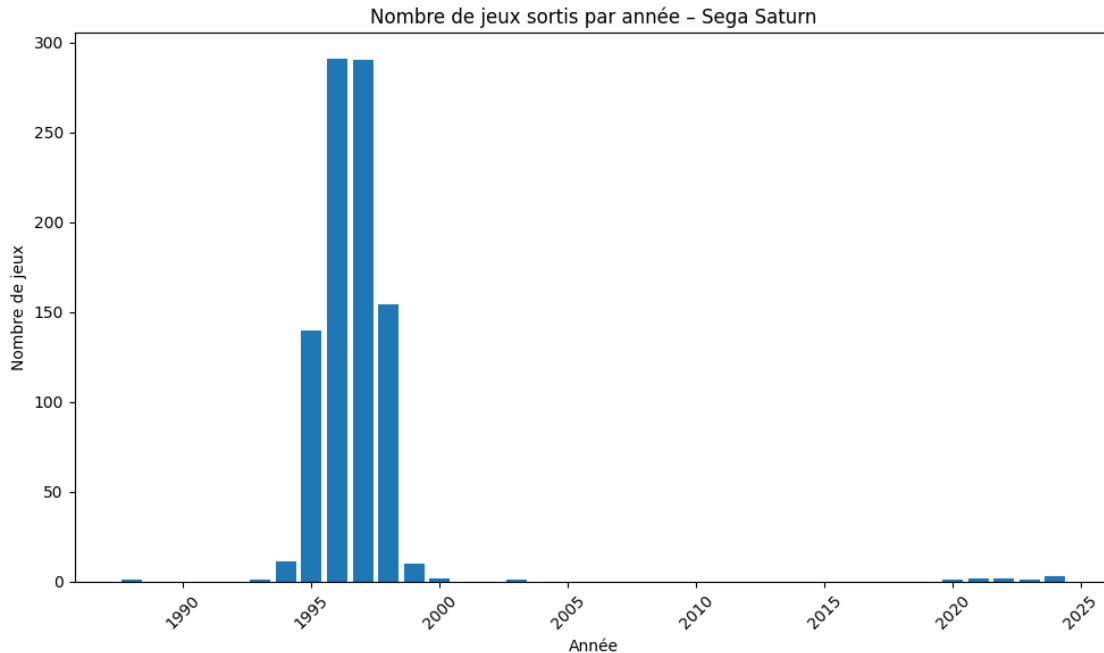


A l'époque, les DLC, remake ou même les modes n'étaient pas populaires voir inexistant, c'est pour ça que sur la Sega Neptune, il y a presque que des "main Game" soit des jeux principaux

```
[174]: # Requête pour comparer Sega avec les autres compagnies sur Sega Saturn
query_comparaison_compagnies = f"""
SELECT
    c.nomCompagnie,
    COUNT(DISTINCT j.idJeu) AS nbJeux
FROM {SCHEMA}.JEU j
JOIN {SCHEMA}.DATESORTIE ds ON ds.idJeu = j.idJeu
JOIN {SCHEMA}.COMPAGNIEJEU cj ON cj.idJeu = j.idJeu
JOIN {SCHEMA}.COMPAGNIE c ON c.idCompagnie = cj.idCompagnie
WHERE ds.idPlateforme = 32
    AND cj.estDeveloppeur = 1
GROUP BY c.nomCompagnie
ORDER BY nbJeux DESC;
"""
```

```
[175]: data_annee = requete_vers_dataframe(
    conn,
    f"""
    SELECT EXTRACT(YEAR FROM ds.dateSortie) AS annee, COUNT(*) AS nbJeux
    FROM {SCHEMA}.DATESORTIE ds
    WHERE ds.idPlateforme = 32
    GROUP BY EXTRACT(YEAR FROM ds.dateSortie)
    ORDER BY annee
    """
)

plt.figure(figsize=(10, 6))
plt.bar(data_annee["ANNEE"], data_annee["NBJEUX"])
plt.xlabel("Année")
plt.ylabel("Nombre de jeux")
plt.title("Nombre de jeux sortis par année - Sega Saturn")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Quand on regarde le graphique des jeux publiés sur cette plateforme, on peut voir que l'année où il y a plus de jeux publiés est 1996, et que pendant une dizaine d'années il n'y a pas eu de publication. Ce qui est intéressant à voir avec ce graphique, c'est qu'il y a des jeux qui ont été publiés vers 2020, ce sont des remakes.

4 2. Comparaison de la Sega Saturn avec d'autres plateformes

4.0.1 Statistiques de base

```
[176]: # Récupérer les données, les transformer si nécessaire, puis afficher sous
      ↪ forme de tableau
data = requete_vers_dataframe(
    conn,
    f"""
    SELECT DISTINCT J.titreJeu, J.scoreIGDB, J.scoreAgregéJeu, J.ScoreJeu
    FROM {SCHEMA}.PLATEFORME P
    INNER JOIN {SCHEMA}.DATESORTIE DS ON P.idPlateforme = DS.idPlateforme
    INNER JOIN {SCHEMA}.JEU J ON J.idJeu = DS.idJeu
    INNER JOIN {SCHEMA}.MODALITEJEU MJ ON MJ.idJeu = J.idJeu
    INNER JOIN {SCHEMA}.MODALITE M ON M.idModalite = MJ.idModalite
    WHERE P.nomPlateforme LIKE :libelle AND J.scoreJeu IS NOT NULL
    ORDER BY J.scoreJeu DESC
    """,
    {"libelle": NOM_PLATEFORME}
)
```

```

fig, ax = plt.subplots(figsize=(30, 100)) # Taille du graphique
data_top_10 = data.head(787)
ax.axis('off')

table1 = ax.table(cellText=data_top_10.values, colLabels=data_top_10.columns,
    ↪loc='center', cellLoc='center', colLoc='center', bbox=[0, 0.19, 1, 0.8])

table1.auto_set_font_size(False)
table1.set_fontsize(12)

# première ligne en gras
for (row, col), cell in table1.get_celld().items():
    if row == 0:
        cell.set_text_props(weight='bold')

plt.title(f"Classement des jeux les mieux notées de la plateforme_
    ↪'{NOM_PLATEFORME}' ", fontweight='bold')
plt.show()

```


Classement des jeux les mieux notés de la plateforme Sega Saturn

TITRE/JEU	SCORE/100	SCORE/100	SCORE/10
Twisted Gears: Let Us Cling Together	88.91691000100462	82.0	80.94849991177751
Sakura Siren	86.03031086322206	nan	86.01833800322206
Wet 17	86.00463266377633	nan	86.00463266377633
Resist: Dragon Rage	87.75623942202090	nan	87.75623942202090
Guardian Heroes	88.04773594999046	85.0	87.327346747454723
Katana Strongman	88.90946377383228	80.0	86.949333009330403
Dragon Force	86.76666376263205	nan	86.76666376263205
Metaphor	86.4733333387337	nan	86.3573333758737
Parade Fighter	85.46589759867328	nan	85.46589759867328
Winning Force II	86.23346398753243	nan	86.23346398753243
Resist: Dragon 3 Zero	85.46519893200006	nan	83.431188933333006
Robotekids	85.33630307333346	nan	85.33630307333346
Lunar 3: Eternal Blue Complete	85.03330000000000	nan	85.03330000000000
Super Puzzle Fighter II Turbo	77.03333333242447	85.0	81.439833306333774
Three Wonders	85.17333667986663	nan	85.17333667986663
DefPact	85.07989753426268	nan	81.167386375554238
Hero's Adventure	86.8546635453333	nan	80.9598335453333
Real Bout Fatal Fury Special	86.73333666666666	nan	86.73333666666666
Warcraft II: The Dark Saga	86.03796388333334	nan	80.03796388333334
Ballistic of 68 88	nan	80.0	80.0
MultiWarrior 2: Brutal Combat Edition	nan	80.0	80.0
DarkFights	76.87733336773669	nan	76.87733336773669
Ninja Lion ST	79.85326385773659	80.0	79.82661382766497
OverClimb: Clumping Donald Duck	76.93333333877498	nan	79.93333333877498
Real Bout Fatal Fury	79.62334663667676	nan	79.62334663667676
DarkBusters 3	78.59466667333388	nan	78.59466667333388
Elevator Action Returns	78.33663333333375	nan	78.33663333333375
Minorsaku Tets: Post Legend	77.88893388897753	nan	77.88893388897753
Rayman	77.47336333338876	77.66666666666667	77.46938406342772
Marvel Super Heroes vs. Street Fighter	76.43873768766957	nan	76.43873768766957
Over Samurais: Soul Hackers	76.96673466638882	75.0	76.246373353333343
Overbl	76.38852233333369	nan	76.04633333333369
The Last Watch: Jurassic Park	75.86266677337399	nan	75.86266677337399
Snatcher	75.17946377424262	nan	75.17946377424262
Loaded	75.25226666362652	75.0	75.12626665966268
Alien vs. Street Fighter	74.86956366667333	nan	74.86956366667333
Highly Infected...	68.44633333333322	87.0	73.05273826380006
Phantom	73.396388897733877	nan	73.396388897733877
Samurai Shodown IV: Amakusa's Revenge	72.86749377338676	nan	72.86749377338676
Indie Wings Special	72.20633333888888	nan	72.20533338888888
Scars II	71.93336333338865	nan	71.93336333338865
The King of Fighters '95	71.73463733883358	nan	71.73463733883358
Night Warriors: DarkBusters: Revenge	70.93873373333347	nan	70.93873373333347
Shinobu Monogatari	70.93266734683876	nan	70.93266734683876
Die Hard Arcade	70.59666666338797	nan	70.59666666338797
Notes Plus	65.337333338873462	80.0	70.339389333348763
Just Cause II: Hollywood	70.47346333338833	nan	70.47346333338833
Fighters Megaplex	70.36334666666663	nan	70.36334666666663
Secret Strike	69.978663366336712	nan	69.978663366336712
Clarius I	69.97833333338826	nan	69.97833333338826
Waka Waka 7	68.93333333338465	nan	68.93333333338465
CardBot	nan	80.0	80.0
Battle Ganga	68.8783988824895	nan	68.8783988824895
Doom	67.44633333883332	70.0	66.73388888366837
Dragon Ball 2: The Legend	67.33888333338806	nan	67.33888333338806
Majral Drop II	67.03388933333307	nan	67.03388933333307
Ugma Battle: The March of the Black Queen	67.33438834633328	nan	67.33438834633328
Marvel Super Heroes	67.27766766666637	nan	67.27766766666637
The King of Fighters '93	70.33333333888888	85.0	66.66663364633309
Planet Robo: Legend of Hideo	66.86673766666676	nan	66.86673766666676
After Burner II	65.94422462388332	nan	65.94422462388332
Castle of Illusion starring Mickey Mouse	75.338233333373366	70.0	63.16666337766666
Dead or Alive	65.23388896347081	nan	64.23388896347081
Virtua Fighter Kids	65.336633338833734	nan	63.336633338833734
Freestyle BattleGP '95	nan	80.0	80.0
Contra: Legacy of War	78.97962666333333	nan	59.97962666333333
Street Fighter: The Movie	58.66333333338826	nan	58.66333333338826
Copier	76.66666666338888	40.0	55.38882338879775
Virtua Fighter	52.24667396633346	nan	52.24667396633346
All Star Baseball '97	nan	50.0	50.0
99% Quarterback Club ST	nan	50.0	50.0
Slackin' Warriors	nan	50.0	50.0
Narrative of the Three Kingdoms to End of the	nan	40.0	40.0
OffRoad: Interceptor Extreme	nan	40.0	40.0
Iron Man and G.I. Warrior II: Heavy Metal	40.0	40.0	40.0
CRASH	nan	30.0	30.0
Bulky 3D	23.07337375888586	nan	23.07337375888586

Cette requête a pour but de récupérer des informations sur les jeux les mieux notés sur une plateforme spécifique, ici la plateforme Sega Saturn. Elle est sur les jeux développés par Sega et extrait des informations détaillées telles que :

- Le titre du jeu
- Le score IGDB
- Le score agrégé
- Le score combiné (scoreJeu)

Tout est classé par ordre décroissant de scoreJeu afin d'avoir les meilleurs jeux en haut du tableau

```
[177]: query = f"""
SELECT
    c.nomCompagnie,
    c.paysCompagnie,
    j.scoreJeu
FROM {SCHEMA}.JEU j
JOIN {SCHEMA}.DATESORTIE ds ON ds.idJeu = j.idJeu
JOIN {SCHEMA}.COMPAGNIEJEU cj ON cj.idJeu = j.idJeu
JOIN {SCHEMA}.COMPAGNIE c ON cj.idCompagnie = c.idCompagnie
WHERE ds.idPlateforme = 32
    AND j.scoreJeu IS NOT NULL
    AND cj.estDeveloppeur = 1
"""

df = requete_vers_dataframe(conn, query)

groupes = {}
for i in range(len(df)):
    compagnie = df.iloc[i]["NOMCOMPAGNIE"]
    pays = df.iloc[i]["PAYSCOMPAGNIE"]
    score = df.iloc[i]["SCOREJEU"]
    key = (compagnie, pays)
    if key not in groupes:
        groupes[key] = []
    groupes[key].append(score)

def calculs_manuels(scores):
    scores.sort()
    n = len(scores)
    moyenne = sum(scores) / n
    variance = sum((x - moyenne)**2 for x in scores) / (n - 1) if n > 1 else 0
    ecart_type = variance ** 0.5
    min_val = scores[0]
    max_val = scores[-1]
    mediane = scores[n//2] if n % 2 else (scores[n//2 - 1] + scores[n//2]) / 2
```

```

q1 = scores[n // 4]
q3 = scores[(3 * n) // 4]
return {
    "moyenne": round(moyenne, 2),
    "écart-type": round(ecart_type, 2),
    "min": min_val,
    "Q1": q1,
    "médiane": mediane,
    "Q3": q3,
    "max": max_val
}

resultats = []
for (compagnie, pays), scores in groupes.items():
    stats = calculs_manuels(scores)
    stats["compagnie"] = compagnie
    stats["pays"] = pays
    resultats.append(stats)

import pandas as pd
df_resultat = pd.DataFrame(resultats)[
    ["compagnie", "pays", "moyenne", "écart-type", "min", "Q1", "médiane",
    ↪ "Q3", "max"]
]
display(df_resultat)

```

	compagnie	pays	moyenne	écart-type	min \
0	Data East	392	74.53	6.00	67.601985
1	Game Arts	392	79.63	5.07	76.048112
2	CAVE Interactive	392	80.57	0.84	79.977289
3	Electronic Arts	840	69.98	0.00	69.976900
4	Sega Enterprises, Ltd.	392	71.54	11.84	63.169263
5	SNK	392	73.08	6.14	66.666883
6	Ubi Pictures	250	77.57	0.00	77.569584
7	Psikyo	None	69.00	0.00	69.000000
8	Yumekobo	392	80.73	0.00	80.732559
9	J-Force	None	86.77	0.00	86.769998
10	Taito	392	72.72	4.76	69.970166
11	Capcom	392	73.31	9.78	55.642320
12	Visual Concepts	840	89.50	0.00	89.504653
13	Konami	392	69.62	20.03	40.000000
14	Sega AM2	392	62.26	6.21	55.249076
15	Raizing	392	68.88	0.00	68.879360
16	Camelot Software Planning	392	81.99	2.46	79.745094
17	Team Andromeda	392	85.64	2.36	83.491707
18	Koei	392	49.00	0.00	49.000000
19	Sonic Team	392	72.94	1.11	71.932842
20	Traveller's Tales	826	71.93	0.00	71.932842

21	SUNSOFT	392	67.18	2.39	65.802708
22	Jaleco Ltd.	None	70.55	0.00	70.553955
23	Burst Studios	None	70.48	0.00	70.477043
24	Quest	392	79.16	16.70	67.354288
25	Neversoft Entertainment	840	50.00	0.00	50.000000
26	Studio Alex	392	83.22	0.00	83.221348
27	Iguana Entertainment	840	50.00	0.00	50.000000
28	Atlus	392	75.60	0.92	74.948980
29	Psygnosis	826	60.00	0.00	60.000000
30	Toaplan	392	86.36	0.00	86.357323
31	Treasure	392	84.53	6.67	70.918367
32	Quantum Factory	None	80.00	0.00	80.000000
33	Riverhillsoft	392	67.35	0.00	67.354288
34	Lobotomy Software	840	73.59	0.00	73.592307
35	iNiS	392	49.00	0.00	49.000000
36	Square Enix	392	90.96	0.00	90.964960
37	Dreamworks Interactive	840	75.44	0.00	75.442000
38	Red Company	392	90.02	0.00	90.019339
39	Tose	392	67.72	0.00	67.724404
40	Funcom Dublin Ltd.	None	50.00	0.00	50.000000
41	Blizzard Entertainment	840	80.01	0.00	80.007563
42	Appaloosa Interactive	348	71.33	8.23	58.979630
43	Video System	392	72.21	0.00	72.205124
44	Atelier Double	392	60.00	0.00	60.000000
45	Team Dead or Alive	None	61.22	0.00	61.221681
46	Real Time	None	38.50	0.00	38.500000
47	G3 Interactive	None	55.31	0.00	55.308025
48	Big Ape Productions	None	80.96	0.00	80.959607
49	Sega Technical Institute	840	70.59	0.00	70.590501
50	BEC	392	67.72	0.00	67.724404
51	SEGA AM1	None	70.59	0.00	70.590501
52	Gremlin Interactive	826	75.13	0.00	75.126300
53	Crystal Dynamics	840	40.00	0.00	40.000000
54	Eidetic	840	23.03	0.00	23.027121
55	Kronos Digital Entertainment	840	30.00	0.00	30.000000
56	Funcom	578	55.31	0.00	55.308025
57	Backbone Entertainment	840	81.42	0.00	81.416853

	Q1	médiane	Q3	max
0	67.601985	77.989519	77.989519	77.989519
1	76.048112	79.634730	83.221348	83.221348
2	79.977289	80.572492	81.167696	81.167696
3	69.976900	69.976900	69.976900	69.976900
4	63.169263	71.544763	79.920263	79.920263
5	66.666883	72.286907	79.822544	80.732559
6	77.569584	77.569584	77.569584	77.569584
7	69.000000	69.000000	69.000000	69.000000
8	80.732559	80.732559	80.732559	80.732559

9	86.769998	86.769998	86.769998	86.769998
10	69.970166	69.970166	78.209263	78.209263
11	70.928714	75.784138	81.416853	85.460098
12	89.504653	89.504653	89.504653	89.504653
13	75.175846	77.587923	83.300764	83.300764
14	55.249076	60.543841	70.062346	70.062346
15	68.879360	68.879360	68.879360	68.879360
16	79.745094	81.988520	84.231946	84.231946
17	83.491707	85.644003	87.796299	87.796299
18	49.000000	49.000000	49.000000	49.000000
19	71.932842	72.942559	73.952277	73.952277
20	71.932842	71.932842	71.932842	71.932842
21	65.802708	65.802708	69.939184	69.939184
22	70.553955	70.553955	70.553955	70.553955
23	70.477043	70.477043	70.477043	70.477043
24	67.354288	79.159624	90.964960	90.964960
25	50.000000	50.000000	50.000000	50.000000
26	83.221348	83.221348	83.221348	83.221348
27	50.000000	50.000000	50.000000	50.000000
28	74.948980	75.598558	76.248137	76.248137
29	60.000000	60.000000	60.000000	60.000000
30	86.357323	86.357323	86.357323	86.357323
31	86.969332	87.327385	87.327385	87.327385
32	80.000000	80.000000	80.000000	80.000000
33	67.354288	67.354288	67.354288	67.354288
34	73.592307	73.592307	73.592307	73.592307
35	49.000000	49.000000	49.000000	49.000000
36	90.964960	90.964960	90.964960	90.964960
37	75.442000	75.442000	75.442000	75.442000
38	90.019339	90.019339	90.019339	90.019339
39	67.724404	67.724404	67.724404	67.724404
40	50.000000	50.000000	50.000000	50.000000
41	80.007563	80.007563	80.007563	80.007563
42	75.442000	75.442000	75.442000	75.442000
43	72.205124	72.205124	72.205124	72.205124
44	60.000000	60.000000	60.000000	60.000000
45	61.221681	61.221681	61.221681	61.221681
46	38.500000	38.500000	38.500000	38.500000
47	55.308025	55.308025	55.308025	55.308025
48	80.959607	80.959607	80.959607	80.959607
49	70.590501	70.590501	70.590501	70.590501
50	67.724404	67.724404	67.724404	67.724404
51	70.590501	70.590501	70.590501	70.590501
52	75.126300	75.126300	75.126300	75.126300
53	40.000000	40.000000	40.000000	40.000000
54	23.027121	23.027121	23.027121	23.027121
55	30.000000	30.000000	30.000000	30.000000
56	55.308025	55.308025	55.308025	55.308025

57 81.416853 81.416853 81.416853 81.416853

```
[178]: top3_compagnies = requete_vers_dataframe(
    conn,
    f"""
    SELECT CJ.idCompagnie, C.nomCompagnie, AVG(J.scoreJeu) AS moyenne_score
    FROM {SCHEMA}.JEU J
    INNER JOIN {SCHEMA}.DATESORTIE DS ON J.idJeu = DS.idJeu
    INNER JOIN {SCHEMA}.PLATEFORME P ON DS.idPlateforme = P.idPlateforme
    INNER JOIN {SCHEMA}.COMPAGNIEJEU CJ ON CJ.idJeu = J.idJeu
    INNER JOIN {SCHEMA}.COMPAGNIE C ON CJ.idCompagnie = C.idCompagnie
    WHERE P.nomPlateforme LIKE :libelle AND J.scoreJeu IS NOT NULL
    AND CJ.estDeveloppeur = 1
    GROUP BY CJ.idCompagnie, C.nomCompagnie
    ORDER BY moyenne_score DESC
    FETCH FIRST 3 ROWS ONLY
    """,
    {"libelle": NOM_PLATEFORME}
)

top3_ids = top3_compagnies["IDCOMPAGNIE"].tolist()

params = {
    "libelle": NOM_PLATEFORME,
    "id1": top3_ids[0],
    "id2": top3_ids[1],
    "id3": top3_ids[2]
}

performance_autres_plateformes = requete_vers_dataframe(
    conn,
    f"""
    SELECT C.nomCompagnie, P.nomPlateforme, AVG(J.scoreJeu) AS moyenne_score
    FROM {SCHEMA}.JEU J
    INNER JOIN {SCHEMA}.DATESORTIE DS ON J.idJeu = DS.idJeu
    INNER JOIN {SCHEMA}.PLATEFORME P ON DS.idPlateforme = P.idPlateforme
    INNER JOIN {SCHEMA}.COMPAGNIEJEU CJ ON CJ.idJeu = J.idJeu
    JOIN {SCHEMA}.COMPAGNIE C ON CJ.idCompagnie = C.idCompagnie
    WHERE CJ.idCompagnie IN (:id1, :id2, :id3)
    AND J.scoreJeu IS NOT NULL
    AND P.nomPlateforme != :libelle
    AND CJ.estDeveloppeur = 1
    GROUP BY C.nomCompagnie, P.nomPlateforme
    ORDER BY C.nomCompagnie, moyenne_score DESC
    """,
    params
)
```

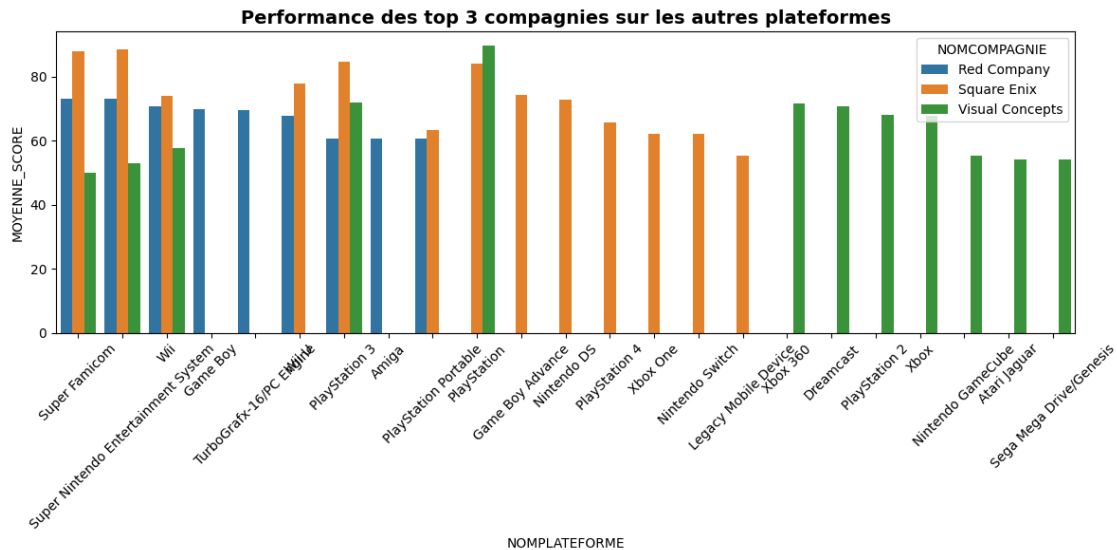
```

print(performance_autres_plateformes.columns)

plt.figure(figsize=(12, 6))
sns.barplot(
    data=performance_autres_plateformes,
    x="NOMPLATEFORME",
    y="MOYENNE_SCORE",
    hue="NOMCOMPAGNIE"
)
plt.title("Performance des top 3 compagnies sur les autres plateformes",
          fontweight='bold',
          fontstyle='italic',
          fontfamily='serif',
          fontcolor='red',
          fontsize=14)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```

```
Index(['NOMCOMPAGNIE', 'NOMPLATEFORME', 'MOYENNE_SCORE'], dtype='object')
```



```

[179]: nombre_jeux_autres_plateformes = requete_vers_dataframe(
    conn,
    f"""
    SELECT C.nomCompagnie, P.nomPlateforme, COUNT(DISTINCT J.idJeu) AS nb_jeux
    FROM {SCHEMA}.JEU J
    INNER JOIN {SCHEMA}.DATESORTIE DS ON J.idJeu = DS.idJeu
    INNER JOIN {SCHEMA}.PLATEFORME P ON DS.idPlateforme = P.idPlateforme
    INNER JOIN {SCHEMA}.COMPAGNIEJEU CJ ON CJ.idJeu = J.idJeu
    INNER JOIN {SCHEMA}.COMPAGNIE C ON CJ.idCompagnie = C.idCompagnie
    WHERE CJ.idCompagnie IN (:id1, :id2, :id3)
    AND P.nomPlateforme != :libelle
    """
)

```

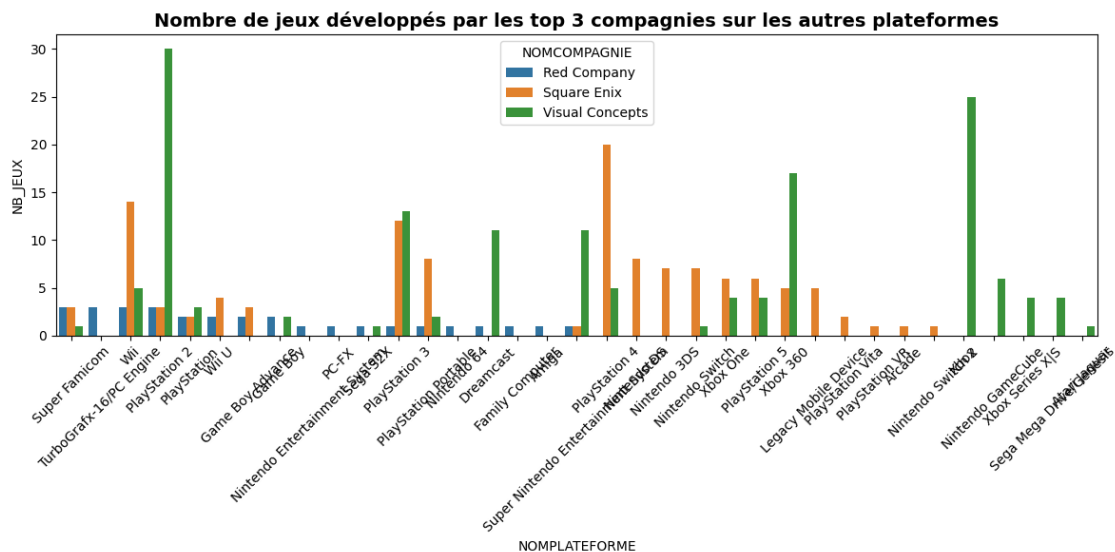
```

        AND CJ.estDeveloppeur = 1
    GROUP BY C.nomCompagnie, P.nomPlateforme
    ORDER BY C.nomCompagnie, nb_jeux DESC
    """,
    params
)

plt.figure(figsize=(12, 6))
sns.barplot(
    data=nombre_jeux_autres_plateformes,
    x="NOMPLATEFORME",    # Adapter si les colonnes sont en majuscules
    y="NB_JEUX",
    hue="NOMCOMPAGNIE"
)

plt.title("Nombre de jeux développés par les top 3 compagnies sur les autres_
↪plateformes", fontsize=14, fontweight='bold')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```



```

[180]: evolution_nb_jeux = requete_vers_dataframe(
    conn,
    f"""
    SELECT EXTRACT(YEAR FROM DS.dateSortie) AS annee, COUNT(DISTINCT J.idJeu)
    ↪AS nb_jeux
    FROM {SCHEMA}.JEU J
    INNER JOIN {SCHEMA}.DATESORTIE DS ON J.idJeu = DS.idJeu
    """
)

```

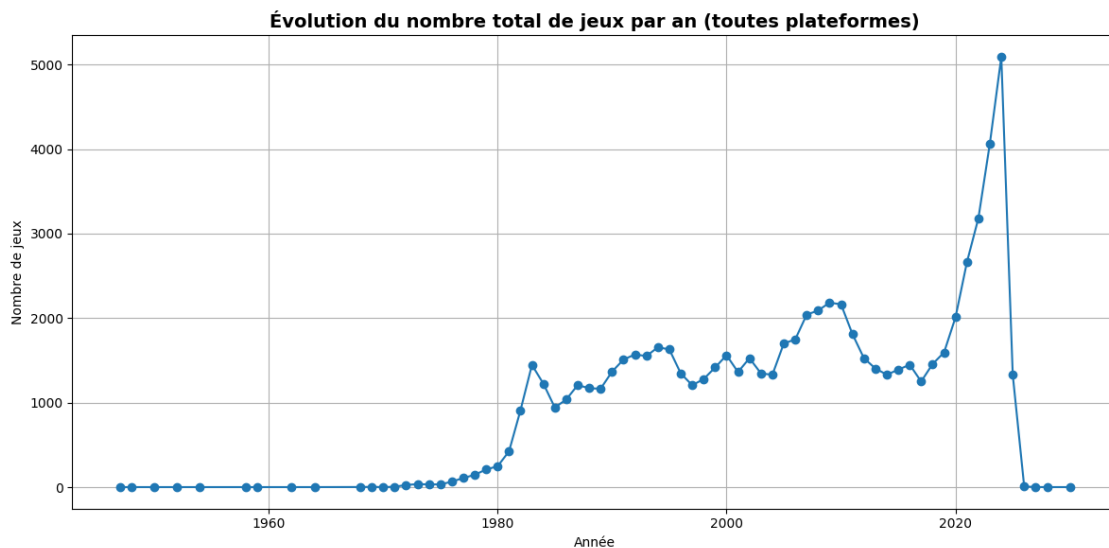


```

WHERE DS.dateSortie IS NOT NULL
GROUP BY EXTRACT(YEAR FROM DS.dateSortie)
ORDER BY annee
"""
)

plt.figure(figsize=(12, 6))
plt.plot(evolution_nb_jeux["ANNEE"], evolution_nb_jeux["NB_JEUX"], marker='o')
plt.title("Évolution du nombre total de jeux par an (toutes plateformes)",
         ↪ fontsize=14, fontweight='bold')
plt.xlabel("Année")
plt.ylabel("Nombre de jeux")
plt.grid(True)
plt.tight_layout()
plt.show()

```



La courbe des jeux vidéo ne fait que augmenter car avant, les jeux vidéo étaient assez niche voir mal vus, mais au fur et à mesure du temps, l'industrie du jeu vidéo s'est démocratisée et est devenue tout public. Cette courbe du nombre de jeux totaux ne va faire que d'augmenter d'année en année car la popularité, elle aussi, ne fait que d'augmenter.

5 3. Contributions de 'Sega Entreprises, Ltd' sur la 'Sega Saturn'

```

[181]: # Calculer avec une requête et afficher le nombre de compagnies ayant développé
         ↪ sur cette plateforme
data = requete_vers_dataframe(
    conn,
    f"""

```

```

SELECT COUNT(DISTINCT CJ.idJeu)
FROM {SCHEMA}.PLATEFORME P
INNER JOIN {SCHEMA}.DATESORTIE DS ON P.idPLATEFORME = DS.IdPlateforme
INNER JOIN {SCHEMA}.COMPAGNIEJEU CJ ON CJ.idJeu = DS.idJeu
INNER JOIN {SCHEMA}.COMPAGNIE C ON C.idcompagnie = CJ.idcompagnie
WHERE P.nomPlateforme LIKE :libelle AND CJ.idCompagnie = 4981
""",
{"libelle": NOM_PLATEFORME}
)

nbJeu = int(data.iloc[0, 0])
print("Nombre de jeux que 'Sega entreprises, Ltd' a développé sur la 'Sega_
↳Saturn': ", nbJeu)

```

Nombre de jeux que 'Sega entreprises, Ltd' a développé sur la 'Sega Saturn': 45

```

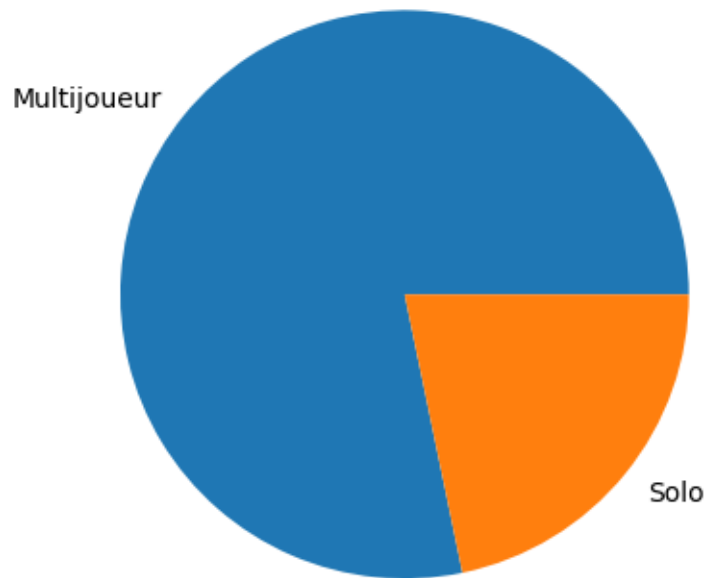
[182]: type_jeux = requete_vers_dataframe(
    conn,
    f"""
SELECT COUNT(M.NomModalite)
FROM {SCHEMA}.COMPAGNIEJEU CJ
INNER JOIN {SCHEMA}.JEU J ON J.idJeu = CJ.idJeu
INNER JOIN {SCHEMA}.MODALITEJEU MJ ON MJ.idJeu = J.idJeu
INNER JOIN {SCHEMA}.MODALITE M ON M.idModalite = MJ.idModalite
INNER JOIN {SCHEMA}.DATESORTIE DS ON J.idJeu = DS.idJeu
INNER JOIN {SCHEMA}.PLATEFORME P ON DS.idPlateforme = P.idPlateforme
WHERE P.nomPlateforme LIKE :libelle AND idCompagnie = 4981 AND M.idModalite_
↳= 2
""",
{"libelle": NOM_PLATEFORME}
)

nbMulti = int(type_jeux.iloc[0, 0])
nbMulti = 100*nbMulti/45
nbSolo = 100*(45-nbMulti)/45
labels = 'Multijoueur', 'Solo'
sizes = [nbMulti, nbSolo]

plt.figure()
plt.title("Répartition entre les jeux Multijoueur et Solo", fontsize=14,
↳fontweight = 'bold')
plt.pie(sizes, labels=labels)
plt.show()

```

Répartition entre les jeux Multijoueur et Solo



On peut voir grâce à ce graphique, que la spécialisation de cette compagnie sont les jeux vidéo multijoueurs

```
[183]: data_genre = requete_vers_dataframe(
    conn,
    f"""
    SELECT g.nomGenre, COUNT(*) AS nbJeux
    FROM {SCHEMA}.JEU j
    INNER JOIN {SCHEMA}.GENREJEU gj ON j.idJeu = gj.idJeu
    INNER JOIN {SCHEMA}.GENRE g ON gj.idGenre = g.idGenre
    INNER JOIN {SCHEMA}.DATESORTIE ds ON ds.idJeu = j.idJeu
    INNER JOIN {SCHEMA}.COMPAGNIEJEU CJ ON CJ.idJeu = J.idJeu
    WHERE ds.idPlateforme = 32 AND idCompagnie = 4981
    GROUP BY g.nomGenre
    ORDER BY nbJeux DESC
    """
)

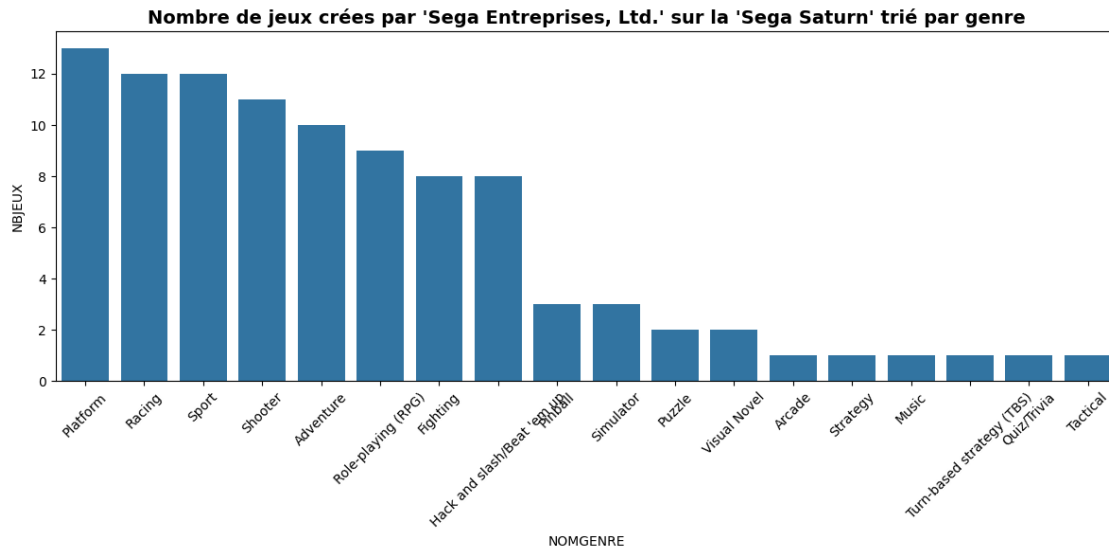
plt.figure(figsize=(12, 6))
sns.barplot(
    data=data_genre,
```

```

x="NOMGENRE",
y="NBJEUX",
)

plt.title("Nombre de jeux créés par 'Sega Entreprises, Ltd.' sur la 'Sega_
↳ Saturn' trié par genre", fontsize=14, fontweight = 'bold')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```



6 4 Comparaison de Sega Entreprises, Ltd avec 3 autres compa- nies (EA SPORT, Quest et Sega) sur la Sega Saturn

```

[184]: compagnie4 = {
    "libelle": NOM_PLATEFORME,
    "id1": 4981,
    "id2": 1827,
    "id3": 112,
    "id4": 454
}

```

```

[185]: # Requête SQL pour récupérer les données des jeux par année et compagnie
TotJeu_top4 = requete_vers_dataframe(
    conn,
    f"""
    SELECT EXTRACT(YEAR FROM DS.dateSortie) AS annee, COUNT(*) AS nbJeux, C.
    ↳ nomCompagnie

```

```

FROM {SCHEMA}.JEU j
INNER JOIN {SCHEMA}.DATESORTIE ds ON ds.idJeu = j.idJeu
INNER JOIN {SCHEMA}.COMPAGNIEJEU CJ ON CJ.idJeu = J.idJeu
INNER JOIN {SCHEMA}.COMPAGNIE C ON C.idCompagnie = CJ.idCompagnie
INNER JOIN {SCHEMA}.PLATEFORME P ON DS.idPlateforme = P.idPlateforme
WHERE P.nomPlateforme = :libelle AND CJ.idCompagnie IN (:id1, :id2, :id3, :
↳id4)
GROUP BY EXTRACT(YEAR FROM DS.dateSortie), C.nomCompagnie
ORDER BY annee
""",
compagnie4
)

# Vérification des colonnes du DataFrame
print(TotJeu_top4.columns)

# Suppression des lignes avec des valeurs manquantes dans les colonnes "ANNEE"
↳ou "NBJEUX"
TotJeu_top4 = TotJeu_top4.dropna(subset=["ANNEE", "NBJEUX"])

# Conversion des colonnes en entier après avoir supprimé les valeurs manquantes
TotJeu_top4["ANNEE"] = TotJeu_top4["ANNEE"].astype(int)
TotJeu_top4["NBJEUX"] = TotJeu_top4["NBJEUX"].astype(int)

# 2. Création du graphique en courbes pour l'évolution des jeux par année pour
↳chaque compagnie
plt.figure(figsize=(12, 6))

# Tracer la courbe pour chaque compagnie
for compagnie in TotJeu_top4["NOMCOMPAGNIE"].unique():
    compagnie_data = TotJeu_top4[TotJeu_top4["NOMCOMPAGNIE"] == compagnie]
    plt.plot(compagnie_data["ANNEE"], compagnie_data["NBJEUX"], marker='o',
↳label=compagnie)

# Ajouter un titre et des labels
plt.title("Évolution du nombre de jeux par année pour 4 compagnies sur la
↳plateforme", fontsize=14, fontweight='bold')
plt.xlabel("Année")
plt.ylabel("Nombre de jeux")
plt.xticks(rotation=45)
plt.grid(True)

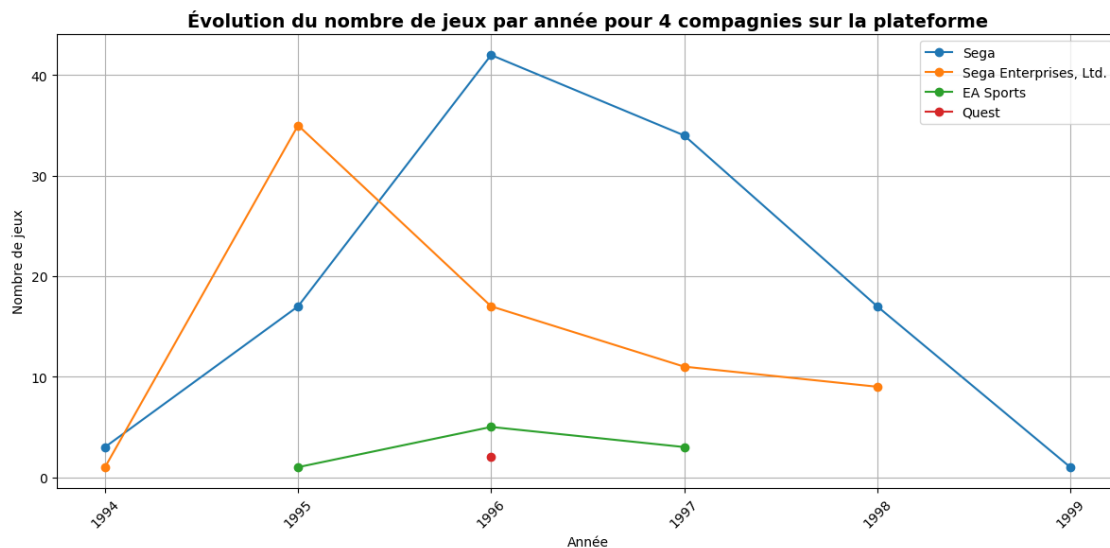
# Ajouter une légende
plt.legend()

# Afficher le graphique
plt.tight_layout()

```

```
plt.show()
```

```
Index(['ANNEE', 'NBJEUX', 'NOMCOMPAGNIE'], dtype='object')
```



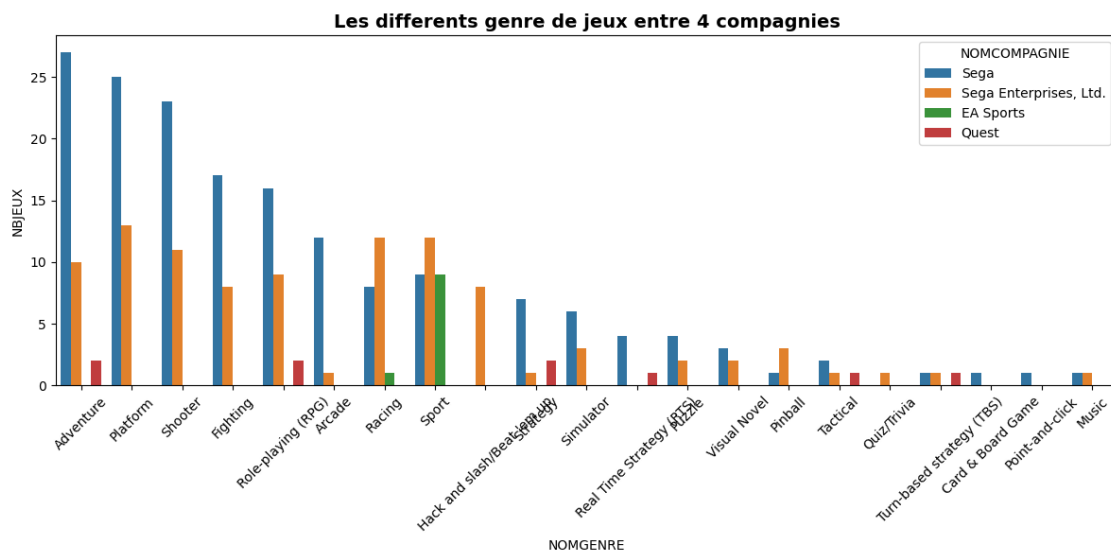
On peut voir avec ce graphique, que Sega et Sega entreprises, Ltd on publiés beaucoup de jeux sur cette console ce qui est normal car c'est leur propre console, alors que Quest et EA Sport, n'ont pas publiés beaucoup de jeux mais ont publié des jeux de qualité car les 2 ont un jeu dans le top 25 des jeux les mieux notés de cette console.

```
[186]: Genre_top4 = requete_vers_dataframe(
    conn,
    f"""
    SELECT g.nomGenre, COUNT(*) AS nbJeux, C.nomCompagnie
    FROM {SCHEMA}.JEU j
    INNER JOIN {SCHEMA}.GENREJEU gj ON j.idJeu = gj.idJeu
    INNER JOIN {SCHEMA}.GENRE g ON gj.idGenre = g.idGenre
    INNER JOIN {SCHEMA}.DATESORTIE ds ON ds.idJeu = j.idJeu
    INNER JOIN {SCHEMA}.COMPAGNIEJEU CJ ON CJ.idJeu = J.idJeu
    INNER JOIN {SCHEMA}.COMPAGNIE C ON C.idCompagnie = CJ.idCompagnie
    INNER JOIN {SCHEMA}.PLATEFORME P ON DS.idPlateforme = P.idPlateforme
    WHERE P.nomPlateforme = :libelle AND CJ.idCompagnie IN (:id1, :id2, :id3, :
↳ id4)
    GROUP BY g.nomGenre, C.nomCompagnie
    ORDER BY nbJeux DESC
    """,
    compagnie4
)

print(Genre_top4.columns)
```

```
plt.figure(figsize=(12, 6))
sns.barplot(
    data=Genre_top4,
    x="NOMGENRE",
    y="NBJEUX",
    hue="NOMCOMPAGNIE"
)
plt.title("Les differents genre de jeux entre 4 compagnies", fontsize=14,
    ↪fontweight='bold')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
Index(['NOMGENRE', 'NBJEUX', 'NOMCOMPAGNIE'], dtype='object')
```



7 5 Evolution de la compagnie

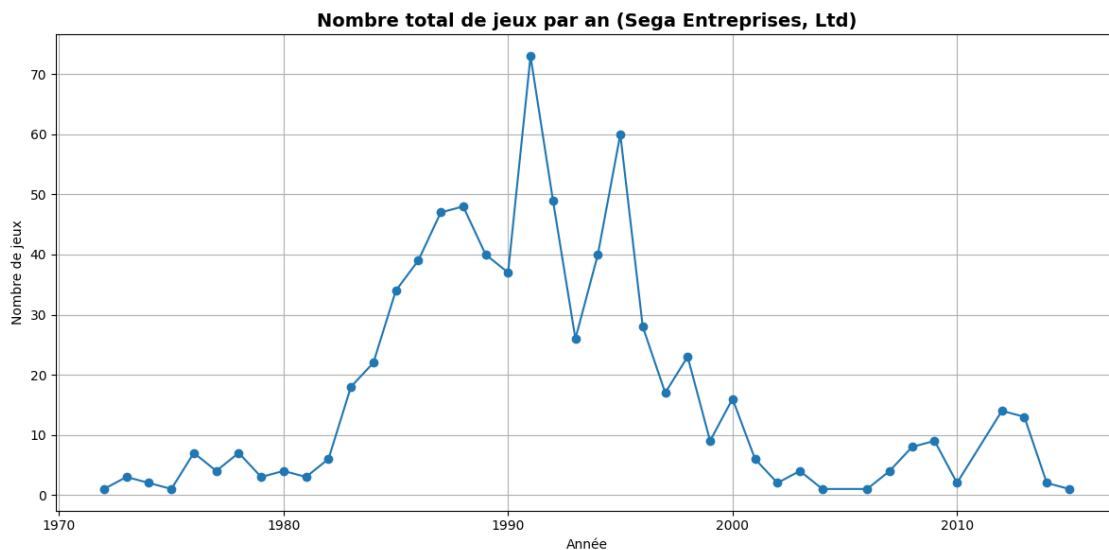
```
[187]: evolution_nb_jeux = requete_vers_dataframe(
    conn,
    f"""
    SELECT EXTRACT(YEAR FROM DS.dateSortie) AS annee, COUNT(DISTINCT J.idJeu)
    ↪AS nb_jeux
    FROM {SCHEMA}.JEU J
    INNER JOIN {SCHEMA}.DATESORTIE DS ON J.idJeu = DS.idJeu
    INNER JOIN {SCHEMA}.COMPAGNIEJEU CJ ON CJ.idJeu = J.idJeu
    INNER JOIN {SCHEMA}.PLATEFORME P ON P.idPlateforme = DS.idPlateforme
    WHERE DS.dateSortie IS NOT NULL AND idCompagnie = 4981
    """
)
```

```

GROUP BY EXTRACT(YEAR FROM DS.dateSortie)
ORDER BY annee
"""
)

plt.figure(figsize=(12, 6))
plt.plot(evolution_nb_jeux["ANNEE"], evolution_nb_jeux["NB_JEUX"], marker='o')
plt.title("Nombre total de jeux par an (Sega Entreprises, Ltd)", fontsize=14,
        fontweight='bold')
plt.xlabel("Année")
plt.ylabel("Nombre de jeux")
plt.grid(True)
plt.tight_layout()
plt.show()

```



On peut voir que cette compagnie existe depuis longtemps, elle ne publiait pas beaucoup de jeux. Puis à partir de 1982, elle n'a fait qu'augmenter jusqu'à arriver à 74 jeux vendus en 1992. Ensuite elle n'a fait que diminuer (à part l'augmentation vers 2010) jusqu'à arriver à 1 jeu publié en 2015. Cette compagnie a donc créé de nombreux jeux mais à présent elle ne produit plus beaucoup, voire plus du tout. Dans le futur, cette entreprise va sûrement ne plus poster de jeux vidéo ou de temps en temps (tous les 2 ans par exemple).