PRAC 2

1. Descripció del dataset. Perquè és important i quina pregunta/problema pretén respondre?.

Aquest dataset es el que vaig obtenir a la practica anterior, consta d'un llistat de videojocs, amb diferent característiques des de 1951 fins a 2020. Tenim un total de 114712 registres, i per cada registre tenim valors possibles a 24 columnes diferents, indicades a continuació:

- GameTittle: Nom del videojoc
- Genre: Gènere del videojoc
- Platform: Plataforma del videojoc (nomes una)
- Platforms**:** Plataformes del videojoc (quan es multiplataforma)
- Released: Data de sortida
- Published by: Editora del videojoc
- Developed by: Desenvolupador del joc
- Add-on: Indica si es un complement per a un videojoc
- Gameplay: Característiques del joc
- Official Site: Nom de la pagina oficial
- Perspective: Perspectiva
- Setting: Ambientació
- Vehicular: Si un joc de vehicles, indica el tipus de vehicle
- Narrative: Tipus de narrativa
- Pacing: Ritme del joc
- Special Edition: Indica si es una edició especial
- ESRB Rating: Classificació per edat ESRB
- Interface: Tipus d'interfície
- Misc: Varies característiques: regional, llicencia...
- Visual: Visualització
- Art: Estil de disseny del joc
- Sport: En els joc esportius, indica el esport
- Educational: En els jocs educatius, indica la branca
- Amazon Rating: Classificació per edat amazon

A continuació indicarem diverses preguntes que podríem respondre després de realitzar el anàlisis de les dades:

- Quina es la plataforma mes longeva?
- Quina es la plataforma a la que se han desenvolupat mes videojocs
- Les consoles mes modernes duren mes anys que les antigues?
- Evolució de la producció de videojocs al llarg dels anys
- Evolució de les consoles de la mateixa marca (ex: playstation, nintendo)

2. Integració i selecció de les dades d'interès a analitzar.

Per aquest dataset, tenim bastants atributs, que no es troben per a la majoria de registres. Com hem pogut veure, tenim registres que no son jocs pròpiament dits, si no que son complements, aquestes registres els eliminarem directament. Per altra banda, eliminarem les columnes que contenguin un major nombre de valors desconeguts, de les quals no podrem treure informació. Com veurem mes endavant, ens quedarem amb les següents columnes: GameTittle, Genre, Platform, Published by, Released, Developed by, Gameplay i Perspective.

3. Neteja de les dades.

En primer lloc realitzem la carrega del dataset i comprovem que s'ha carregat correctament.

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
data = pd.read_csv('gamesDataSet_original.csv')
data.head()
```

```
C:\Users\recurso\.conda\envs\Master\lib\site-packages\IPython\core\interactiveshell.py:3063: DtypeWarning: Columns (23) have mi
xed types.Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Out[2]:

	Add-on	GameTittle	Genre	Platforms	Published by	Released	Platform	Developed by	Gameplay	Official Site	...	Pacing	Special Edition	E Ra
0	Customization / outfit / skin	Dead Space: Pedestrian Pack	DLC / add-on	PlayStation 3, Xbox 360	Electronic Arts, Inc.	Nov 13, 2008	NaN	NaN	NaN	NaN	...	NaN	NaN	
1	Customization / outfit / skin, Item	Dead Space: Scorpion Pack	Compilation, DLC / add-on	PlayStation 3, Xbox 360	Electronic Arts, Inc.	Nov 13, 2008	NaN	NaN	NaN	NaN	...	NaN	NaN	
2	Customization / outfit / skin	Dead Space: Scorpion Suit	DLC / add-on	PlayStation 3, Xbox 360	Electronic Arts, Inc.	Oct 09, 2008	NaN	NaN	NaN	NaN	...	NaN	NaN	
		Dead Space:		PlayStation										

A continuació procedim a selecciones les files dels registres que no son complements de videojocs, per eliminar aquests del dataset. També fem un recompte, mostrant el percentatge de registres que tenen la columna buida, d'aquesta manera, identifica'm les columnes que ens podrien ser útils.

```
In [25]: data = data[data['Add-on'].isnull()]
         data.isnull().sum()  / data.shape[0] * 100

Out[25]: Add-on            100.000000
         GameTittle          0.001127
         Genre               0.075521
         Platforms          57.196479
         Published by        7.589301
         Released            0.001127
         Platform           42.805776
         Developed by       24.775409
         Gameplay           35.691018
         Official Site      78.544135
         Perspective        21.868413
         Setting            65.796860
         Vehicular          90.158594
         Narrative          88.064294
         Pacing             83.246728
         Special Edition    97.235028
         ESRB Rating        89.509339
         Interface          75.461298
         Misc               87.663018
         Visual             67.619509
         Art                91.313953
         Sport              93.365420
         Educational        96.832625
         Amazon Rating      99.997746
         dtype: float64
```

En la seguen captura, podrem veure com ja seleccionem les columnes que ens interessen. Com es pot veure, tenim dues columnes de plataformes, una que nomes posa una, i en una altra, es una llista. Per a solucionar aquest problema, he comprovat que totes les files que tenen Platform rellentat, tenen Platforms a nul. D'aquesta manera he pogut traspassar la columna Platform a Platforms, i he eliminat la columna, d'aquesta manera nomes tenim una columna per indicar la plataforma o plataformes a las que esta disponible el joc.

```
In [4]: data = data[['GameTittle','Genre','Platforms','Published by','Released','Platform','Developed by','Gameplay','Perspective']]

In [5]: platformNN = data[~data['Platform'].isnull()].shape[0]
        platformNNandPlatformsN = data[~data['Platform'].isnull() & data['Platforms'].isnull()].shape[0]

        print (platformNN,platformNNandPlatformsN)

        50741 50741

In [6]: data.loc[~data['Platform'].isnull(), 'Platforms'] = data.loc[~data['Platform'].isnull(), 'Platform']
        data = data.drop(['Platform'], axis=1)
        data.isnull().sum()

Out[6]: GameTittle          1
        Genre              67
        Platforms           2
        Published by     6733
        Released            1
        Developed by    21980
        Gameplay        31664
        Perspective     19401
        dtype: int64
```

3.1. Les dades contenen zeros o elements buits? Com gestionaries aquests casos?

Dada la naturalesa del nostre dataset, estem limitats a la hora d'utilitzar tècniques per omplir les dades dels elements buits. Tenim 3 atributs amb pocs elements buits, aquests els tractarem manualment eliminant-los del dataset.

```
In [7]: data[data['GameTittle'].isnull()]
```

Out[7]:

	GameTittle	Genre	Platforms	Published by	Released	Developed by	Gameplay	Perspective
68205	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
In [9]: data = data[~data['GameTittle'].isnull()]
        data = data[~data['Platforms'].isnull()]
        data.isnull().sum()

Out[9]: GameTittle          0
        Genre              66
        Platforms           0
        Published by     6731
        Released            0
        Developed by    21978
        Gameplay        31663
        Perspective     19400
        dtype: int64
```

A continuació, mirarem si tenim valors 'Unknown'. Com podem veure tenim 3, al atribut Released. Com que es una xifra molt petita, també els eliminem del dataset. A continuació, omplim totes les dades del dataset que estaven buides amb el valor 'Unknown'. Com podem veure ens hem quedat amb un dataset amb files i 8 columnes

```
In [10]: (data=='Unknown').sum()

Out[10]: GameTittle      0
         Genre           0
         Platforms       0
         Published by    0
         Released        3
         Developed by    0
         Gameplay        0
         Perspective     0
         dtype: int64
```

```
In [12]: data = data[data['Released'] != 'Unknown']
         data = data.fillna('Unknown')
```

```
In [13]: data.shape

Out[13]: (88712, 8)
```

3.2. Identificació i tractament de valors extrems.

Com que els atributs nos tots cadenes de text, no tenim problemes de valors extrems. En l'únic camp que podríem intentar mirar-lo, seria en el camp released, encara que volem conservar els valors extrems, que lo mes probable es que es trobin en els primers anys, quan la quantitat de videojocs era mes baixa que en la actualitat.

3.3. Retocs finals

Lo primer que faré, es modificar el camp released, per indicar nomes l'any de sortida del videojoc, no la data sencera. Tenim registres que ja tenen l'any, així que creem una funció que retorna els quatre darrers dígits, si la longitud es superior a 4, i la apliquem a tots els registres. D'aquesta manera ens queda nomes l'any com podem veure a continuació.

```
In [13]: data.Released
```

```
Out[13]: 5         May 09, 2012
         7         May 01, 2012
         8         Apr 05, 2016
         9         Feb 14, 2014
         10        May 13, 2015
                       ...
         114701    May 01, 2012
         114703            2009
         114704    Nov 13, 2008
         114706    Oct 13, 2010
         114711    Oct 27, 2011
         Name: Released, Length: 88712, dtype: object
```

```
In [14]: def getYearIfCompleteDate(date):
             if len(date) > 4:
                 date = date[-4:]
             return date
```

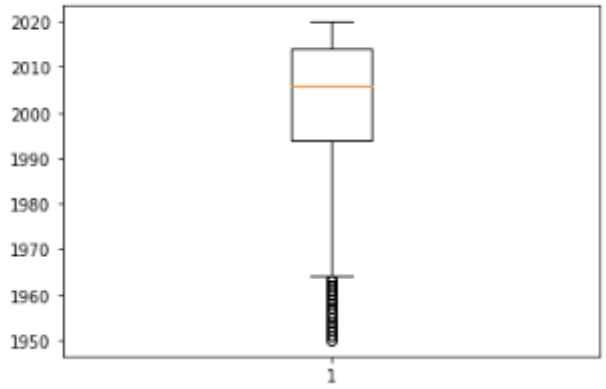
```
In [15]: data.Released = data.Released.apply(getYearIfCompleteDate)
         data.Released.unique()
```

Com havien esbrinat abans, en el boxplot podem veure com el valor extrem es concentren als primers anys de la historia dels videojocs.

```
Out[15]: array(['2012', '2016', '2014', '2015', '2009', '2010', '2004', '2002',
                '2005', '2019', '2013', '1986', '2003', '1994', '2006', '2011',
                '2007', '2008', '2017', '2000', '2018', '2020', '1996', '1990',
                '1991', '1989', '1988', '1983', '1997', '1992', '1995', '1980',
                '1987', '1976', '1984', '1982', '1985', '2001', '1999', '1981',
                '1978', '1993', '1998', '1979', '1975', '1974', '1977', '1972',
                '1964', '1965', '1970', '1973', '1951', '1969', '1966', '1968',
                '1971', '1955', '1963', '1961', '1958', '1962', '1959', '1952',
                '1967', '1954', '1956', '1957', '1950', '1953', '1960'],
               dtype=object)
```

```
In [16]: plt.boxplot(pd.to_numeric(data.Released))
```

```
Out[16]: {'whiskers': [<matplotlib.lines.Line2D at 0x244734a3978>,
           <matplotlib.lines.Line2D at 0x244762ecda0>],
          'caps': [<matplotlib.lines.Line2D at 0x244762ece80>,
           <matplotlib.lines.Line2D at 0x2447630a470>],
          'boxes': [<matplotlib.lines.Line2D at 0x244762ec6a0>],
          'medians': [<matplotlib.lines.Line2D at 0x2447630a7b8>],
          'fliers': [<matplotlib.lines.Line2D at 0x2447630ab00>],
          'means': []}
```



El darrer retoc que realitzem al dataset es separar l'atribut platforms, per a que no sigui una llista. Lo que farem es transforma 1 registre que diu que es de mes d'una plataforma, es un registre per plataforma. Com podem comprovar a les imatges següent, lo que en primer lloc es un registre, el numero 8, que es per dos plataformes, se ha transformat en dos registres, el 2 i el 4 n la segona imatge.

```
: data.head()
```

:

	GameTittle	Genre	Platforms	Published by	Released	Developed by	Gameplay	Perspective
5	Dead Space: Super Bundle	Compilation	PlayStation 3	Electronic Arts, Inc.	2012	Unknown	Unknown	Unknown
7	Dead Space: Ultimate Edition	Compilation	PlayStation 3	Electronic Arts, Inc.	2012	Unknown	Unknown	Unknown
8	Dead Star	Action, Strategy / tactics	PlayStation 4, Windows	Armature Studio	2016	Armature Studio	Shooter	Top-down
9	Dead State	Role-Playing (RPG), Strategy / tactics	Windows	DoubleBear Productions	2014	DoubleBear Productions	Unknown	Bird's-eye view
10	Dead State: Reanimated	Role-Playing (RPG), Strategy / tactics	Windows	DoubleBear Productions	2015	DoubleBear Productions	Unknown	Bird's-eye view

```
In [17]: lst_col = 'Platforms'
         data = data.assign(**{lst_col:data[lst_col].str.split(',')})
```

```
In [20]: data = pd.DataFrame({
             col:np.repeat(data[col].values, data[lst_col].str.len())
             for col in data.columns.difference([lst_col])
         }).assign(**{lst_col:np.concatenate(data[lst_col].values)})[data.columns.tolist()]
```

```
In [21]: data = data.rename(columns={"Platforms": "Platform"})
         data.head()
```

Out[21]:

	GameTittle	Genre	Platform	Published by	Released	Developed by	Gameplay	Perspective
0	Dead Space: Super Bundle	Compilation	PlayStation 3	Electronic Arts, Inc.	2012	Unknown	Unknown	Unknown
1	Dead Space: Ultimate Edition	Compilation	PlayStation 3	Electronic Arts, Inc.	2012	Unknown	Unknown	Unknown
2	Dead Star	Action, Strategy / tactics	PlayStation 4	Armature Studio	2016	Armature Studio	Shooter	Top-down
3	Dead Star	Action, Strategy / tactics	Windows	Armature Studio	2016	Armature Studio	Shooter	Top-down
4	Dead State	Role-Playing (RPG), Strategy / tactics	Windows	DoubleBear Productions	2014	DoubleBear Productions	Unknown	Bird's-eye view

```
In [23]: data.shape
Out[23]: (180672, 8)
```

Finalitzant tota la neteja de les dades, ens han quedat 180672 videojocs amb 8 atributs. Canviem el tipus de dades de la data a numèric, i quitem espais en blanc de la columna plataforma. I per acabar, guardem el fitxer com gamesDataSet_final.csv.

```
In [89]: data.Platform  = data.Platform.apply(lambda x: x.strip())
         data.Released = pd.to_numeric(data.Released)

         data.to_csv ('gamesDataSet_final.csv', index = False, header=True)
```

4. Anàlisi de les dades.

4.1. Selecció dels grups de dades que es volen analitzar/comparar (planificació dels anàlisis a aplicar).

Per a realitzar el meu anàlisis i simplificar un poc el gràfics, he seleccionat nomes les plataformes que tenen mes de 1000 videojocs.

```
In [58]: platformList = data.Platform.value_counts()
         platformList = platformList[platformList > 1000]
         print(platformList)
         platformList = platformList[platformList > 1000].index.tolist()
         data2 = data[data.Platform.isin(platformList)]
```

```
Windows            36310
Macintosh          11902
iPhone              7901
iPad                7416
DOS                 7088
Android             6673
PlayStation 4       6086
Linux               5023
PlayStation 3       4996
Commodore 64        4848
Xbox One            4073
Nintendo Switch     3823
Xbox 360            3819
Amiga               3633
ZX Spectrum         3248
PlayStation 2       3149
Browser             2837
Arcade              2628
PlayStation         2486
Wii                 2416
PS Vita             2412
Atari ST            2404
Amstrad CPC         2108
PSP                 2040
Apple II            1962
Nintendo DS         1878
Windows 3.x         1504
Nintendo 3DS        1501
Atari 8-bit         1498
Wii U               1346
NES                 1333
PC-98               1293
MSX                 1209
SNES                1174
Name: Platform, dtype: int64
```

4.2. Comprovació de la normalitat i homogeneïtat de la variància.

Coneixia els mètodes per a realitzar aquestes proves amb variables numèriques, però crec que no es pot fer amb variables categòriques. I no he tingut temps de canviar el dataset.

4.3. Aplicació de proves estadístiques per comparar els grups de dades. En funció de les dades i de l'objectiu de l'estudi, aplicar proves de contrast d'hipòtesis, correlacions, regressions, etc. Aplicar almenys tres mètodes d'anàlisi diferents.

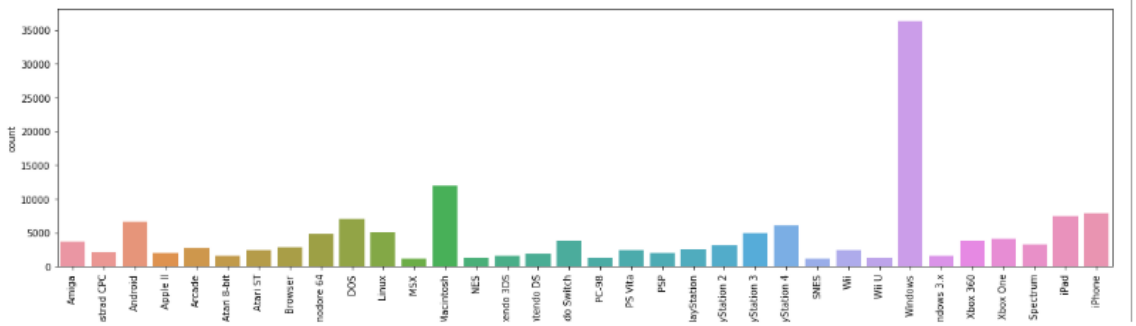
Falta de temps.

5. Representació dels resultats a partir de taules i gràfiques.

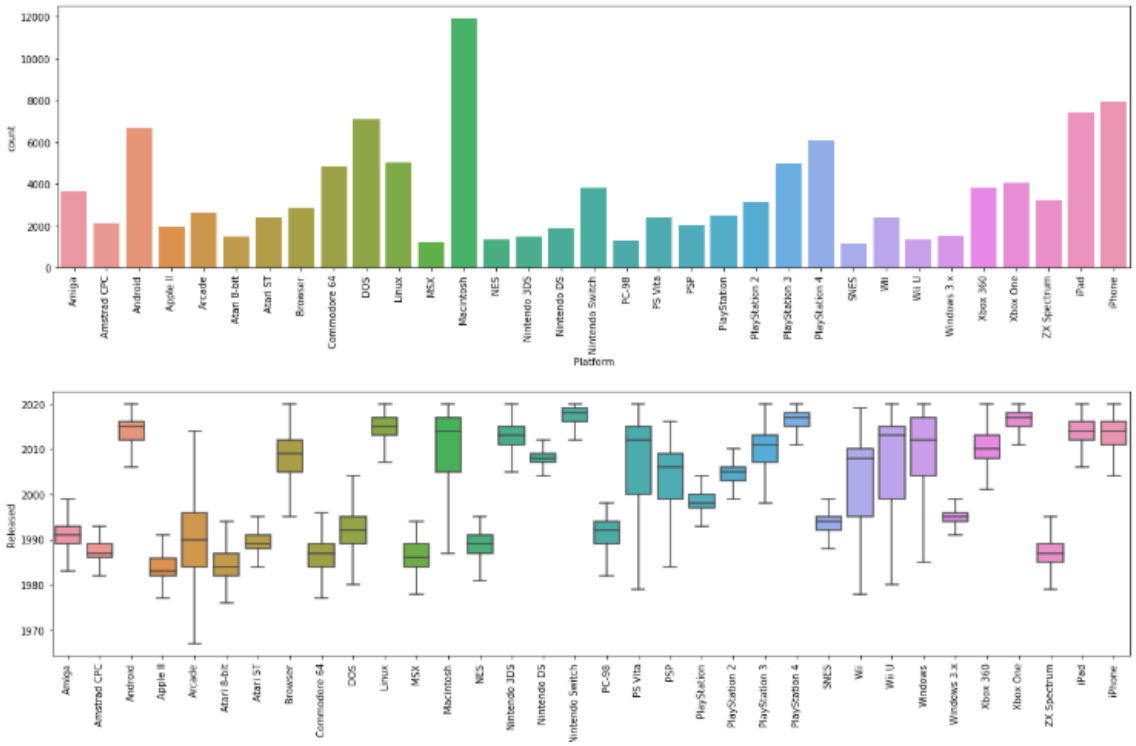
En primer lloc he creat uns gràfics per veure com de productiva es cada plataforma, evidentment, Windows destaca, por lo que la treure, i tornaré a fer la gràfica per apreciar mes les diferencies de la resta de plataformes.

```
varName = 'Platform'

print(data2[varName].nunique())

data2 = data2.sort_values(by=varName, ascending=True)

plt.figure(figsize=(20,5))
plt.xticks(rotation='vertical')
sns.countplot(data = data2, x = varName)
plt.show()

plt.figure(figsize=(20,5))
plt.xticks(rotation='vertical')
sns.countplot(data = data2[data2.Platform != 'Windows'], x = varName)
plt.show()

plt.figure(figsize=(20,5))
plt.xticks(rotation='vertical')
sns.boxplot(data = data2, x=varName, y='Released',showfliers=False)
plt.show()
```

```
34
```

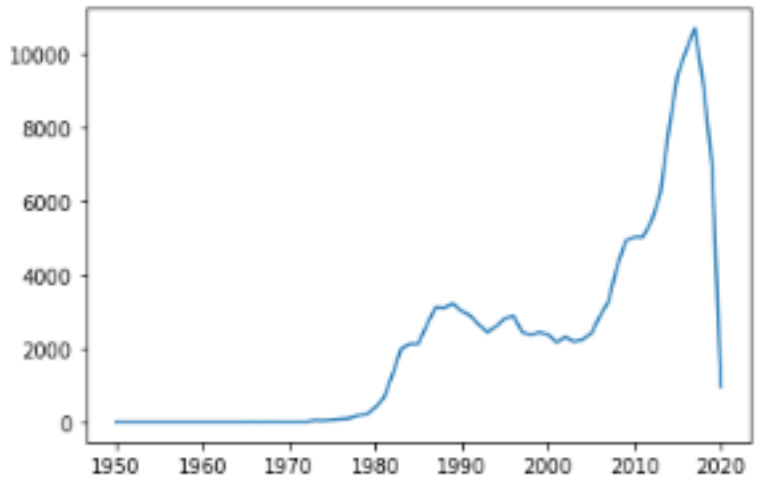


En aquestes gràfiques podem observar com les llevat Windows i mac, els dispositius mòbils son els que tenen mes producció de videojocs. També cal destacar que la maquina àrcade es la que te mes duració.



En la següent gràfica, podem veure l'evolució en la producció de videojocs al llarg dels anys, i com, a partir del 2005 se ha començat a disparar, degut a la facilitat de les noves tecnologies per a la producció de videojocs.

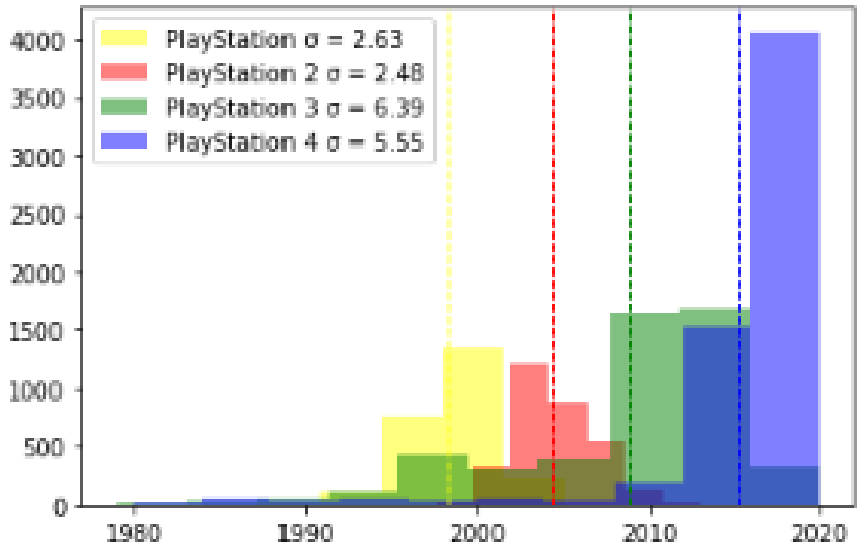
```
In [64]: plt.plot(data2['Released'].value_counts().sort_index())
Out[64]: [<matplotlib.lines.Line2D at 0x203e187d160>]
```

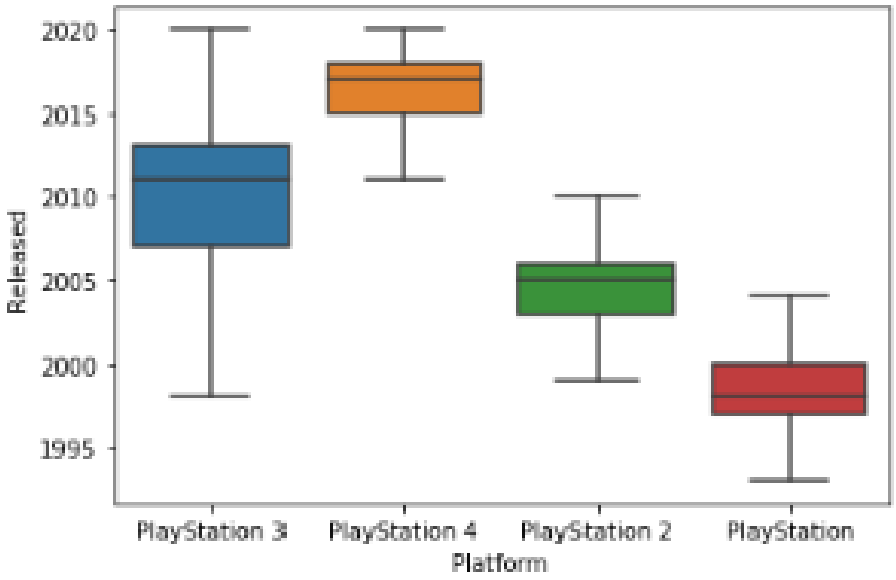


Aquí traurem unes gràfiques per observar l'evo9lucio de les consoles de sobremesa de sony, de la playstation 1 a la 4. Podem observar con de la 1 cada cop es generàvem mes jocs, fins que va sortir la 2. La 2 per la seva complexitat de programació, cada cop es produïen menys videojocs. La 3, va superar les seves dues germanes des de un bon començament, i ja la 4, se ha disparat moltíssim. Això es com he comentat, gracies a que en la actualitat existeixen moltes eines de producció, i molta mes gent dedicada a aquest negoci. Gracies a aquestes gràfiques he pogut detectar un problema, si observem podem veure jocs de la 4 abans de que es produís la consola. Això es degut, a que se ha posat de moda, afegir jocs emulats de consoles anteriors. Per això tenim un joc de la ps4 que ha sigut llançat al 1990. En el meu cas, he acceptat aquests jocs, però es podrien llevar filtrant cada plataforma pel seu any de llançament.

```
In [26]: dataPlays = data2[data2.Platform.str.contains('PlayStation')]
         dataPlays.Platform.value_counts()
Out[26]: PlayStation 4    6086
         PlayStation 3    4996
         PlayStation 2    3149
         PlayStation      2486
         Name: Platform, dtype: int64
```

```
In [56]: def createPlots ():
             plays = ['PlayStation','PlayStation 2', 'PlayStation 3','PlayStation 4']
             colors = ['yellow', 'red', 'green','blue']
             colorIndex = 0

             for target in plays:
                 x = dataPlays[dataPlays['Platform']==target].Released
                 legend = target
                 legend = '%s σ = %s' % (target, round(x.std(),2))
                 plt.hist(x, facecolor=colors[colorIndex], alpha=0.5, label = legend)
                 plt.axvline(x.mean(), color=colors[colorIndex], linestyle='dashed', linewidth=1)
                 colorIndex=(colorIndex+1) % 4
             plt.legend()
             plt.show()

             sns.boxplot(data = dataPlays, x=varName, y='Released',showfliers=False)

         createPlaystationPlots()
```



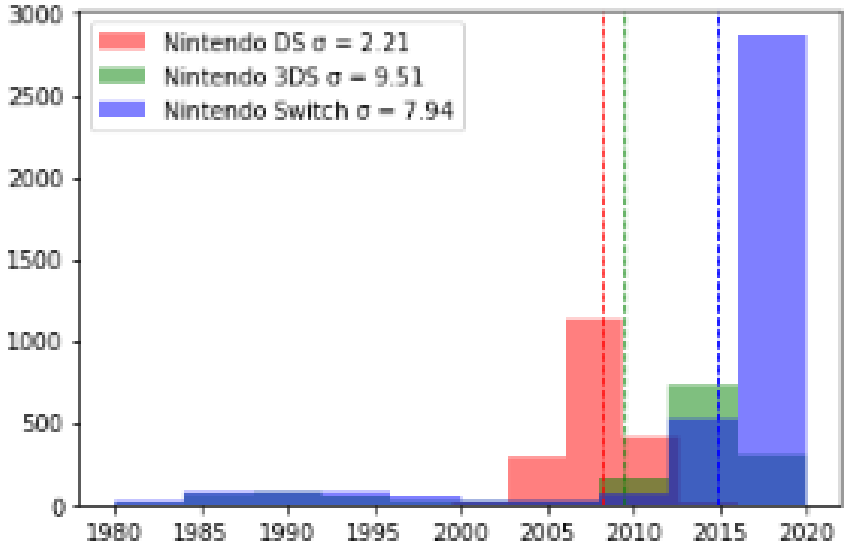


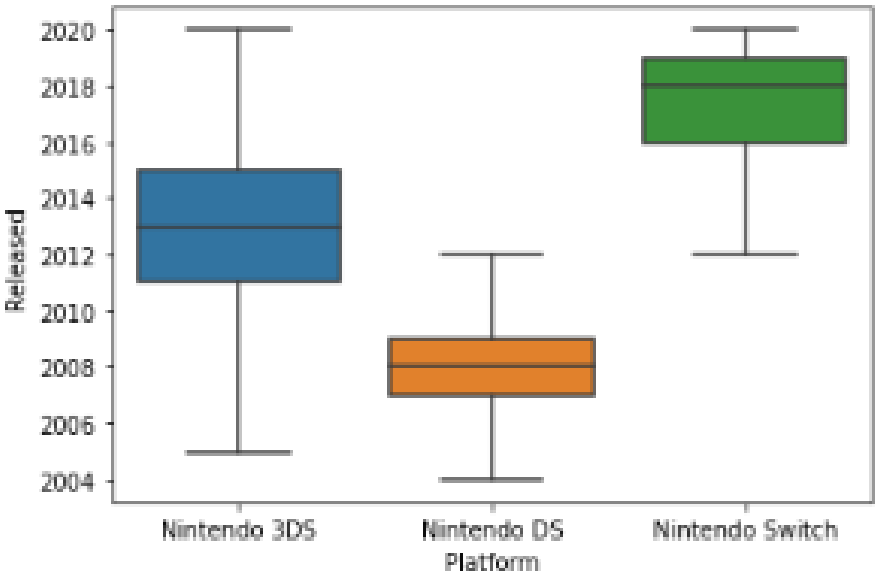
In [49]:
```
dataNintendos = data2[data2.Platform.str.contains('Nintendo')]
dataNintendos.Platform.value_counts()
```

Out[49]:
```
Nintendo Switch    3823
Nintendo DS        1878
Nintendo 3DS       1501
Name: Platform, dtype: int64
```

In [55]:
```
def createNintendoPlots ():
    plays = ['Nintendo DS','Nintendo 3DS', 'Nintendo Switch']
    colors = ['red', 'green','blue']
    colorIndex = 0

    for target in plays:
        x = dataNintendos[dataNintendos['Platform']==target].Released
        legend = target
        legend = '%s σ = %s' % (target, round(x.std(),2))
        plt.hist(x, facecolor=colors[colorIndex], alpha=0.5, label = legend)
        plt.axvline(x.mean(), color=colors[colorIndex], linestyle='dashed', linewidth=1)
        colorIndex=(colorIndex+1) % 3
    plt.legend()
    plt.show()

    sns.boxplot(data = dataNintendos, x=varName, y='Released',showfliers=False)

createNintendoPlots()
```

Per últim tenim les mateixes gràfiques, però amb les consoles portàtils de nintendo DS, 3DS i Switch. Es pot observar com la 3DS, no va ser un gran èxit, però en canvi la Switch a triplicat a les seves progenitores.





6. Resolució del problema. A partir dels resultats obtinguts, quines són les conclusions? Els resultats permeten respondre al problema?

Tornant a les preguntes exposades inicialment, procediré a donar-les resposta:

- Quina es la plataforma mes longeva?
  La maquina d'àrcade
- Quina es la plataforma a la que se han desenvolupat mes videojocs
  Windows
- Les consoles mes modernes duren mes anys que les antigues?
  Ara que se ha assentat mes el mercat i no hi ha tanta lluita i varietat de consoles, duren mes que abans

- Evolució de la producció de videojocs al llarg dels anys
  Entre el 1985 i el 2005 es va trobar mes o menys estable, però a partir del 2005 se ha disparat la producció.
- Evolució de les consoles de la mateixa marca (ex: playstation, nintendo)
  Lo normal es que durin mes i produeixin mes jocs, però es pot donar el cas de que hagin intentat afegir una novetat, i no hagi tingut èxit. Encara que no tenim dades en aquest dataset, l'èxit d'una plataforma pot estar relacionada amb el mercat. Per exemple la psvita en Europa a sigut un fracàs, però en Japó un èxit i es produeixin molt mes jocs per aquest mercat.

7. Codi: Cal adjuntar el codi, preferiblement en R, amb el que s'ha realitzat la neteja, anàlisi i representació de les dades. Si ho preferiu, també podeu treballar en Python.

   El codi font es pot trobar en el fitxer: fmorenobo_TCVD_PRAC2.ipynb