Analise de Filmes Brasileiros e Estrangeiros Exibidos

Baseado nos dados da Ancine, informações por semana 2009 a 2019

URL: https://oca.ancine.gov.br/sites/default/files/repositorio/csv/2122.csv (https://oca.ancine.gov.br/sites/default/files/repositorio/csv/2122.csv)

Willer Gomes Junior

In [1]:
```
# Importar bibliotecas
# por convencao vamos chamar o pandas de pd
import pandas as pd
# Configurando o matplotlib
%matplotlib inline
```

In [2]:
```
# Criar o Dataframe com base no arquivo Excel
# precisamos pular as duas primeiras linhas e as 19 ultimas que não tem infor
macoes corretas
df =  pd.read_excel("2122.xlsx", skiprows=2, skipfooter=19,  encoding = "cp12
52")

df.head()
```

Out[2]:

	Ano de exibição	Semana de exibição	CPB/ROE	Título da obra	Gênero	País(es) produtor(es) da obra	Nacionalidade da obra	Data de Lançamento	Dist
0	2009	semana 01	E1402431200000	007 Quantum of Solace	Ficção	Estados Unidos, Inglaterra	Estrangeira	2008-11-07 00:00:00	
1	2009	semana 01	B0700678800000	5 Frações de uma Quase História	Ficção	Brasil	Brasileira	2008-05-09 00:00:00	Us
2	2009	semana 01	E1600541500000	A Alegria de Emma	Ficção	Alemanha	Estrangeira	2008-10-03 00:00:00	
3	2009	semana 01	E1600594000000	A Bela Junie	Ficção	França	Estrangeira	2009-01-01 00:00:00	
4	2009	semana 01	E1600127600000	A Casa das Coelhinhas	Ficção	Estados Unidos	Estrangeira	2008-10-10 00:00:00	

In [3]:
```
df.tail(20)
```

Out[3]:

	Ano de exibição	Semana de exibição	CPB/ROE	Título da obra	Gênero	País(es) produtor(es) da obra	Nacionalidade da obra	Dat Lançame
61114	2019	semana 52	E1900267700000	O Rei Leão	Ficção	Estados Unidos	Estrangeira	2019-0 00:0
61115	2019	semana 52	E1900483800000	O Reino Gelado: A Terra Dos Espelhos	Animação	Rússia	Estrangeira	2019-1 00:0
61116	2019	semana 52	E1900304200000	O Relatório	Ficção	Estados Unidos	Estrangeira	2019-1 00:0
61117	2019	semana 52	E1900268100000	O Último Amor De Casanova	Ficção	França	Estrangeira	2019-1 00:0
61118	2019	semana 52	B1900108800000	Os Jovens Baumann	Ficção	Brasil	Brasileira	2019-0 00:0
61119	2019	semana 52	B1900487100000	Os Parças 2	Ficção	Brasil	Brasileira	2019-1 00:0
61120	2019	semana 52	E1900458900000	Papicha	Ficção	Argélia, Bélgica, França, Qatar	Estrangeira	2019-1 00:0
61121	2019	semana 52	E1900453900000	Parasita	Ficção	Coréia do Sul	Estrangeira	2019-1 00:0
61122	2019	semana 52	E1900162100000	Pets: A Vida Secreta Dos Bichos 2	Ficção	Estados Unidos, França, Japão	Estrangeira	2019-0 00:0
61123	2019	semana 52	E1900127400000	Playmobil - O Filme	Animação	Alemanha, França	Estrangeira	2019-1 00:0
61124	2019	semana 52	E1900302600000	Rainha De Copas	Ficção	Dinamarca, Suécia	Estrangeira	2019-0 00:0
61125	2019	semana 52	E1900499200000	Retablo	Ficção	Alemanha, Noruega, Peru	Estrangeira	2019-1 00:0
61126	2019	semana 52	E1900358900000	Star Wars: A Ascensão Skywalker	Ficção	Estados Unidos	Estrangeira	2019-1 00:0
61127	2019	semana 52	E1900525400000	Synonyms	Ficção	Alemanha, França, Israel	Estrangeira	2019-1 00:0
61128	2019	semana 52	E1900229100000	Toy Story 4	Animação	Estados Unidos	Estrangeira	2019-0 00:0

	Ano de exibição	Semana de exibição	CPB/ROE	Título da obra	Gênero	País(es) produtor(es) da obra	Nacionalidade da obra	Dat Lançam
61129	2019	semana 52	E1900431700000	Um Amante Francês	Ficção	França	Estrangeira	2019-1 00:0
61130	2019	semana 52	E1900466900000	Um Dia De Chuva Em Nova York	Ficção	Estados Unidos	Estrangeira	2019-1 00:0
61131	2019	semana 52	B1800552500000	Um Dia Para Susana	Documentário	Brasil	Brasileira	2019-1 00:0
61132	2019	semana 52	B1900177100000	Uma	Documentário	Brasil, Índia	Brasileira	2019-1 00:0
61133	2019	semana 52	E1900436800000	Uma Segunda Chance Para Amar	Ficção	Estados Unidos, Reino Unido	Estrangeira	2019-1 00:0

In [4]:
```
df.dtypes
```

Out[4]:
```
Ano de exibição                          int64
Semana de exibição                      object
CPB/ROE                                 object
Título da obra                          object
Gênero                                  object
País(es) produtor(es) da obra           object
Nacionalidade da obra                   object
Data de Lançamento                      object
Distribuidora                           object
Origem da empresa distribuidora         object
Número de salas na semana dos dados      int64
Público na semana dos dados              int64
Renda (R$) na semana dos dados         float64
dtype: object
```

In [5]:
```
# O campo 'Data de Lançamento' não esta como datetime e sim como objeto
# deve haver algum lixo
# converter o campo 'Data de Lançamento' em data
df['Data de Lançamento'] = pd.to_datetime(df['Data de Lançamento'])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~/.local/lib/python3.5/site-packages/pandas/core/arrays/datetimes.py in objec
ts_to_datetime64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, a
llow_object)
   1978         try:
-> 1979             values, tz_parsed = conversion.datetime_to_datetime64(dat
a)
   1980             # If tzaware, these values represent unix timestamps, so
we

pandas/_libs/tslibs/conversion.pyx in pandas._libs.tslibs.conversion.datetime
_to_datetime64()

TypeError: Unrecognized value type: <class 'str'>

During handling of the above exception, another exception occurred:

ParserError                               Traceback (most recent call last)
<ipython-input-5-7b4eb6d221d7> in <module>
      2 # deve haver algum lixo
      3 # converter o campo 'Data de Lançamento' em data
----> 4 df['Data de Lançamento'] = pd.to_datetime(df['Data de Lançamento'])

~/.local/lib/python3.5/site-packages/pandas/util/_decorators.py in wrapper(*a
rgs, **kwargs)
    206                 else:
    207                     kwargs[new_arg_name] = new_arg_value
--> 208             return func(*args, **kwargs)
    209
    210         return wrapper

~/.local/lib/python3.5/site-packages/pandas/core/tools/datetimes.py in to_dat
etime(arg, errors, dayfirst, yearfirst, utc, box, format, exact, unit, infer_
datetime_format, origin, cache)
    772                 result = result.tz_localize(tz)
    773     elif isinstance(arg, ABCSeries):
--> 774         cache_array = _maybe_cache(arg, format, cache, convert_listli
ke)
    775         if not cache_array.empty:
    776             result = arg.map(cache_array)

~/.local/lib/python3.5/site-packages/pandas/core/tools/datetimes.py in _maybe
_cache(arg, format, cache, convert_listlike)
    154         unique_dates = unique(arg)
    155         if len(unique_dates) < len(arg):
--> 156             cache_dates = convert_listlike(unique_dates, True, forma
t)
    157             cache_array = Series(cache_dates, index=unique_dates)
    158     return cache_array

~/.local/lib/python3.5/site-packages/pandas/core/tools/datetimes.py in _conve
rt_listlike_datetimes(arg, box, format, name, tz, unit, errors, infer_datetim
e_format, dayfirst, yearfirst, exact)
    461             errors=errors,
    462             require_iso8601=require_iso8601,
--> 463             allow_object=True,
    464         )
    465

~/.local/lib/python3.5/site-packages/pandas/core/arrays/datetimes.py in objec
ts_to_datetime64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, a
llow_object)
```

```
   1982             return values.view("i8"), tz_parsed
   1983         except (ValueError, TypeError):
-> 1984             raise e
   1985
   1986     if tz_parsed is not None:

~/.local/lib/python3.5/site-packages/pandas/core/arrays/datetimes.py in objec
ts_to_datetime64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, a
llow_object)
   1973             dayfirst=dayfirst,
   1974             yearfirst=yearfirst,
-> 1975             require_iso8601=require_iso8601,
   1976         )
   1977     except ValueError as e:

pandas/_libs/tslib.pyx in pandas._libs.tslib.array_to_datetime()

pandas/_libs/tslib.pyx in pandas._libs.tslib.array_to_datetime()

pandas/_libs/tslib.pyx in pandas._libs.tslib.array_to_datetime_object()

pandas/_libs/tslib.pyx in pandas._libs.tslib.array_to_datetime_object()

pandas/_libs/tslibs/parsing.pyx in pandas._libs.tslibs.parsing.parse_datetime
_string()

~/.local/lib/python3.5/site-packages/dateutil/parser/_parser.py in parse(time
str, parserinfo, **kwargs)
   1366         return parser(parserinfo).parse(timestr, **kwargs)
   1367     else:
-> 1368         return DEFAULTPARSER.parse(timestr, **kwargs)
   1369
   1370

~/.local/lib/python3.5/site-packages/dateutil/parser/_parser.py in parse(sel
f, timestr, default, ignoretz, tzinfos, **kwargs)
    641
    642         if res is None:
--> 643             raise ParserError("Unknown string format: %s", timestr)
    644
    645         if len(res) == 0:

ParserError: Unknown string format: Relançamento
```

Identificando o erro acima 'ParserError: Unknown string format: Relançamento'

In [8]:
```
# Tentando identificar o problema da 'Data de Lançamento'
# Listando todas as 'Data de Lançamento' para analise
df['Data de Lançamento'].value_counts()
```

Out[8]:
```
Relançamento           650
2014-10-23 00:00:00    250
2012-12-21 00:00:00    233
2009-06-05 00:00:00    198
2015-08-20 00:00:00    197
                      ...
2009-09-22 00:00:00      1
2004-11-06 00:00:00      1
2019-02-08 00:00:00      1
2008-02-15 00:00:00      1
2014-11-08 00:00:00      1
Name: Data de Lançamento, Length: 879, dtype: int64
```

In [9]:
```
# São 650 'Data de Lançamento' com o valor 'Relançamento'
```

In [10]:
```
# Alguns filmes estão com data = Relançamento
df[df['Data de Lançamento'] == 'Relançamento'].head()
```

Out[10]:

	Ano de exibição	Semana de exibição	CPB/ROE	Título da obra	Gênero	País(es) produtor(es) da obra	Nacionalidade da obra	Data de Lançamento
519	2009	semana 05	E1600589800000	Os Contos de Canterbury	Ficção	Itália	Estrangeira	Relançamento
636	2009	semana 06	E1600589800000	Os Contos de Canterbury	Ficção	Itália	Estrangeira	Relançamento
746	2009	semana 07	E1600589800000	Os Contos de Canterbury	Ficção	Itália	Estrangeira	Relançamento
847	2009	semana 08	E1600589800000	Os Contos de Canterbury	Ficção	Itália	Estrangeira	Relançamento
956	2009	semana 09	E1600589800000	Os Contos de Canterbury	Ficção	Itália	Estrangeira	Relançamento

Decisão sobre dados incorretos/faltantes/divergentes

In [11]:
```
df.shape
```

Out[11]:
```
(61134, 13)
```

In [12]:
```
# Criar um novo datafrme sem os relançamentos
df_novo = df[df['Data de Lançamento'] != 'Relançamento'].copy()
```

In [13]:
```
df_novo.shape
```

Out[13]:
```
(60484, 13)
```

In [14]:
```
# Vamos verificar os tipos das colunas
df_novo.dtypes
```

Out[14]:
```
Ano de exibição                          int64
Semana de exibição                      object
CPB/ROE                                 object
Título da obra                          object
Gênero                                  object
País(es) produtor(es) da obra           object
Nacionalidade da obra                   object
Data de Lançamento                      object
Distribuidora                           object
Origem da empresa distribuidora         object
Número de salas na semana dos dados      int64
Público na semana dos dados              int64
Renda (R$) na semana dos dados         float64
dtype: object
```

In [15]:
```
# Vamos tentar converter o campo novamente agora sem os Relançamentos no novo dataframe
df_novo['Data de Lançamento'] = pd.to_datetime(df_novo['Data de Lançamento'])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~/.local/lib/python3.5/site-packages/pandas/core/arrays/datetimes.py in objec
ts_to_datetime64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, a
llow_object)
   1978         try:
-> 1979             values, tz_parsed = conversion.datetime_to_datetime64(dat
a)
   1980             # If tzaware, these values represent unix timestamps, so
we

pandas/_libs/tslibs/conversion.pyx in pandas._libs.tslibs.conversion.datetime
_to_datetime64()

TypeError: Unrecognized value type: <class 'str'>

During handling of the above exception, another exception occurred:

ParserError                               Traceback (most recent call last)
<ipython-input-15-aa3d61382203> in <module>
      1 # Vamos tentar converter o campo novamente agora sem os Relançamentos
no novo dataframe
----> 2 df_novo['Data de Lançamento'] = pd.to_datetime(df_novo['Data de Lança
mento'])

~/.local/lib/python3.5/site-packages/pandas/util/_decorators.py in wrapper(*a
rgs, **kwargs)
    206                 else:
    207                     kwargs[new_arg_name] = new_arg_value
--> 208             return func(*args, **kwargs)
    209
    210         return wrapper

~/.local/lib/python3.5/site-packages/pandas/core/tools/datetimes.py in to_dat
etime(arg, errors, dayfirst, yearfirst, utc, box, format, exact, unit, infer_
datetime_format, origin, cache)
    772                 result = result.tz_localize(tz)
    773     elif isinstance(arg, ABCSeries):
--> 774         cache_array = _maybe_cache(arg, format, cache, convert_listli
ke)
    775         if not cache_array.empty:
    776             result = arg.map(cache_array)

~/.local/lib/python3.5/site-packages/pandas/core/tools/datetimes.py in _maybe
_cache(arg, format, cache, convert_listlike)
    154         unique_dates = unique(arg)
    155         if len(unique_dates) < len(arg):
--> 156             cache_dates = convert_listlike(unique_dates, True, forma
t)
    157             cache_array = Series(cache_dates, index=unique_dates)
    158     return cache_array

~/.local/lib/python3.5/site-packages/pandas/core/tools/datetimes.py in _conve
rt_listlike_datetimes(arg, box, format, name, tz, unit, errors, infer_datetim
e_format, dayfirst, yearfirst, exact)
    461             errors=errors,
    462             require_iso8601=require_iso8601,
--> 463             allow_object=True,
    464         )
    465

~/.local/lib/python3.5/site-packages/pandas/core/arrays/datetimes.py in objec
ts_to_datetime64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, a
```

```
llow_object)
   1982             return values.view("i8"), tz_parsed
   1983         except (ValueError, TypeError):
-> 1984             raise e
   1985
   1986     if tz_parsed is not None:

~/.local/lib/python3.5/site-packages/pandas/core/arrays/datetimes.py in objec
ts_to_datetime64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, a
llow_object)
   1973             dayfirst=dayfirst,
   1974             yearfirst=yearfirst,
-> 1975             require_iso8601=require_iso8601,
   1976         )
   1977     except ValueError as e:

pandas/_libs/tslib.pyx in pandas._libs.tslib.array_to_datetime()

pandas/_libs/tslib.pyx in pandas._libs.tslib.array_to_datetime()

pandas/_libs/tslib.pyx in pandas._libs.tslib.array_to_datetime_object()

pandas/_libs/tslib.pyx in pandas._libs.tslib.array_to_datetime_object()

pandas/_libs/tslibs/parsing.pyx in pandas._libs.tslibs.parsing.parse_datetime
_string()

~/.local/lib/python3.5/site-packages/dateutil/parser/_parser.py in parse(time
str, parserinfo, **kwargs)
   1366         return parser(parserinfo).parse(timestr, **kwargs)
   1367     else:
-> 1368         return DEFAULTPARSER.parse(timestr, **kwargs)
   1369
   1370

~/.local/lib/python3.5/site-packages/dateutil/parser/_parser.py in parse(sel
f, timestr, default, ignoretz, tzinfos, **kwargs)
    641
    642         if res is None:
--> 643             raise ParserError("Unknown string format: %s", timestr)
    644
    645         if len(res) == 0:

ParserError: Unknown string format: relançamento
```

In [16]:
```
# Criar um novo dataframe sem os relançamentos
df_novo = df[(df['Data de Lançamento'] != 'Relançamento') & (df['Data de Lanç
amento'] != 'relançamento')].copy()
```

In [17]:
```
# Vamos tentar converter o campo novamente agora sem os Relançamentos no novo
dataframe
df_novo['Data de Lançamento'] = pd.to_datetime(df_novo['Data de Lançamento'])
```

In [18]:
```
df.shape
```

Out[18]:
```
(61134, 13)
```

In [19]:
```
# Vamos verificar os tipos das colunas
df_novo.dtypes
```

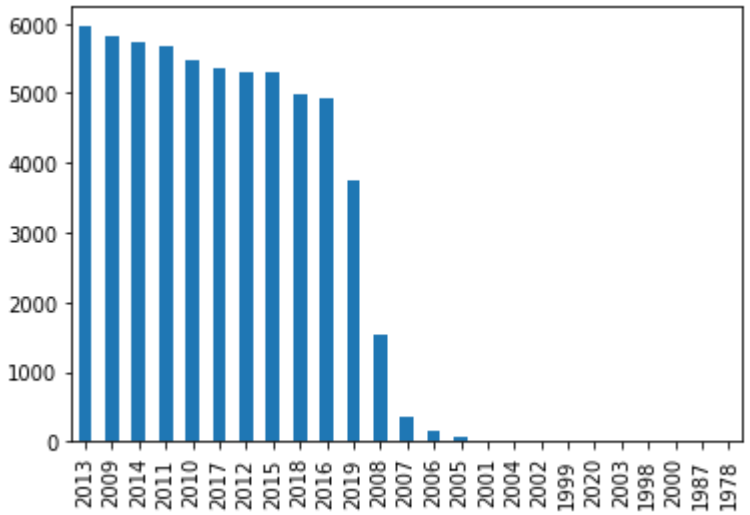
Out[19]:
```
Ano de exibição                              int64
Semana de exibição                          object
CPB/ROE                                     object
Título da obra                              object
Gênero                                      object
País(es) produtor(es) da obra               object
Nacionalidade da obra                       object
Data de Lançamento                  datetime64[ns]
Distribuidora                               object
Origem da empresa distribuidora             object
Número de salas na semana dos dados          int64
Público na semana dos dados                  int64
Renda (R$) na semana dos dados             float64
dtype: object
```

In [20]:
```
# Quais os anos tiveram mais filmes lançados
df_novo['Data de Lançamento'].dt.year.value_counts()
```

Out[20]:
```
2013    5957
2009    5822
2014    5733
2011    5669
2010    5484
2017    5354
2012    5313
2015    5297
2018    4996
2016    4929
2019    3744
2008    1536
2007     341
2006     163
2005      55
2001      22
2004      19
2002      13
1999      10
2020       9
2003       6
1998       5
2000       4
1987       1
1978       1
Name: Data de Lançamento, dtype: int64
```

In [21]:
```
# Vamos plotar um gráfico
df_novo['Data de Lançamento'].dt.year.value_counts().plot.bar()
```

Out[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xaafbd66c>
```



In [22]:
```
# Qual filme com maior bilheteria
df_novo[df_novo['Renda (R$) na semana dos dados'] == df_novo['Renda (R$) na semana dos dados'].max()]
```

Out[22]:

	Ano de exibição	Semana de exibição	CPB/ROE	Título da obra	Gênero	País(es) produtor(es) da obra	Nacionalidade da obra	Data de Lançamento
58087	2019	semana 17	E1900107800000	Vingadores: Ultimato	Ficção	Estados Unidos	Estrangeira	2019-04-25

In [23]:
```
df_novo.nlargest(3, 'Renda (R$) na semana dos dados')
```

Out[23]:

	Ano de exibição	Semana de exibição	CPB/ROE	Título da obra	Gênero	País(es) produtor(es) da obra	Nacionalidade da obra	Data Lançame
58087	2019	semana 17	E1900107800000	Vingadores: Ultimato	Ficção	Estados Unidos	Estrangeira	2019-04
52787	2018	semana 18	E1800067100000	VINGADORES: GUERRA INFINITA	Ficção	Estados Unidos	Estrangeira	2018-04
59027	2019	semana 29	E1900267700000	O Rei Leão	Ficção	Estados Unidos	Estrangeira	2019-07

In [24]:
```
# Quantos filmes brasileiros e estrangeiros
df_novo['Nacionalidade da obra'].value_counts()
```

Out[24]:
```
Estrangeira    44549
Brasileira     15934
Name: Nacionalidade da obra, dtype: int64
```

In [ ]: