??????????

2017 年 12 月 5 日

1 项目：可视化电影数据

对原始数据进行整理，使用 Tableau 创建可视化视图，并用这些视图来对问题进行描述。

1.1 数据整理

首先对拿到的原始数据的字段进行观察分析：

字段名	含义	字段值类型	字段取值分类
id	唯一编号	数值	离散
imdb_id	IMDB 编号	文本	离散
popularity	页面浏览次数	数值	连续/离散 (视考察角度不同可有不同解读)
budget	预算	数值	连续/离散
revenue	票房	数值	连续/离散
original_title	名称	文本	离散
cast	演职人员	文本	离散
homepage	影片主页	文本	离散
director	导演	文本	离散
tagline	宣传语	文本	离散
keywords	关键词/标签	文本	离散
overview	剧情简介	文本	离散
runtime	片长 (分钟数)	数值	连续/离散
genres	类型	文本	离散
production_companies	制作方	文本	离散
release_date	上映日期	文本 (日期)	离散
vote_count	评分次数	数值	连续/离散
vote_average	平均得分	数值	连续/离散
release_year	发行年份	文本 (年)	离散

字段名	含义	字段值类型	字段取值分类
budget_adj	修正预算	数值	连续/离散
revenue_adj	修正票房	数值	连续/离散

原始数据中显然有一些字段与我们想要考察的问题无关，因此我们先根据问题对字段进行一下筛选：* 首先 Id 字段作为总体索引保留 * 考察电影类型是如何随时代变化而变化的，这一问题可通过随着发行年代上映的各个类型的电影数量来考察，因此我们需要保留的是 genres 和 release_year 字段 * 考察环球影业和派拉蒙的电影之间数据指标的差异，差异可能存在于很多字段之中，这里我认为电影公司最关注的应该是盈利的差异，因此我们需要通过 production_companies 、release_year、revenue 和 budget 字段 * 要考察小说改编电影和非小说改编电影的表现有何不同，在上个问题中我们已经考察了票房方面，这里我们可以关注一下关注度和口碑的不同，因此我们保留 keywords、vote_count 和 vote_average 字段 * 自选问题，我们试着探讨一下电影取得高票房是否一定要高投入。

1.1.1 数据清洗

首先我们先对每个字段单独进行特征分析，对异常/缺失值进行处理。

首先针对连续型的数据，我们用 pandas 的 describe 函数和箱线图来进行分析。

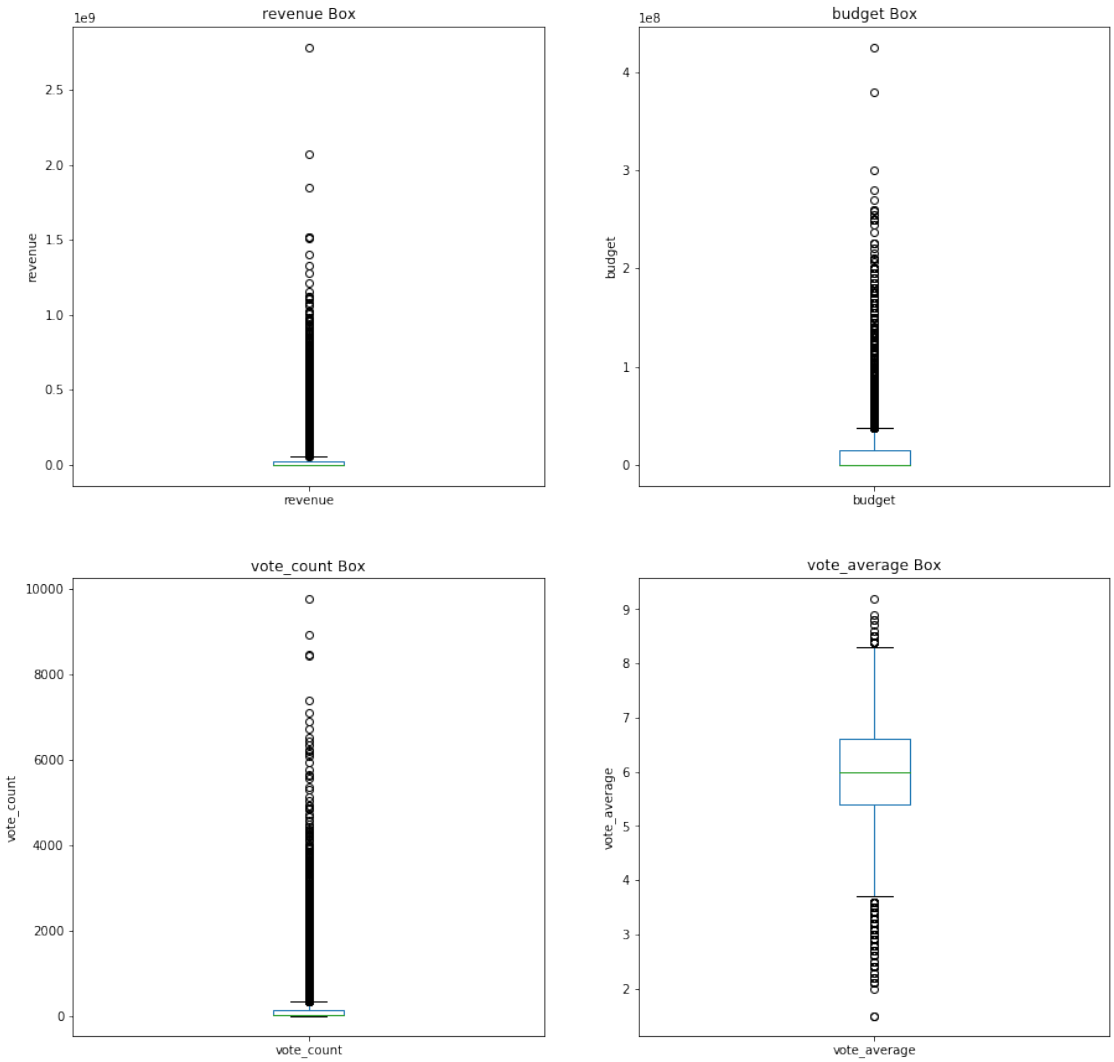
In [1]:
```
# -*- coding: utf-8 -*-

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

df_all = pd.read_csv('movies.csv')
pd.options.display.float_format = '{:,.2f}'.format
```

In [2]:
```
fig,axes=plt.subplots(2,2)

# 展示连续字段的基本统计描述信息
def show_describe_and_boxplot(column,axis):
    df_all[column].plot(figsize=(15,15),kind='box',title=column + ' Box',ax=axis)
    axis.set_ylabel(column)

    print(df_all[column].describe())

show_describe_and_boxplot('revenue',axes[0][0])
```

```
        show_describe_and_boxplot('budget',axes[0][1])
        show_describe_and_boxplot('vote_count',axes[1][0])
        show_describe_and_boxplot('vote_average',axes[1][1])

        plt.show()
```

```
count          10,866.00
mean       39,823,319.79
std       117,003,486.58
min                 0.00
25%                 0.00
50%                 0.00
75%        24,000,000.00
max     2,781,505,847.00
Name: revenue, dtype: float64
count        10,866.00
mean     14,625,701.09
std      30,913,213.83
min              0.00
25%              0.00
50%              0.00
75%      15,000,000.00
max     425,000,000.00
Name: budget, dtype: float64
count   10,866.00
mean       217.39
std        575.62
min         10.00
25%         17.00
50%         38.00
75%        145.75
max      9,767.00
Name: vote_count, dtype: float64
count   10,866.00
mean         5.97
std          0.94
min          1.50
```

```
25%          5.40
50%          6.00
75%          6.60
max          9.20
Name: vote_average, dtype: float64
```



从图形和 describe 信息来看，4 个字段都没有丢失值，vote_count 和 vote_average 字段的分布图形虽然显示有异常值，但经对比判断，都在正常范围，但 revenue 和 budget 字段的数据指标有些异常。经查发现这 2 个字段存在大量值为 0 的数据，我们需要当做缺失值对待，这里采用按年度取对应字段的中位数填充的方法处理，虽然准确度比较低，但我们的考察重点是差异，这种方式至少不会扩大差异性。最后将修改后的值写入文件。

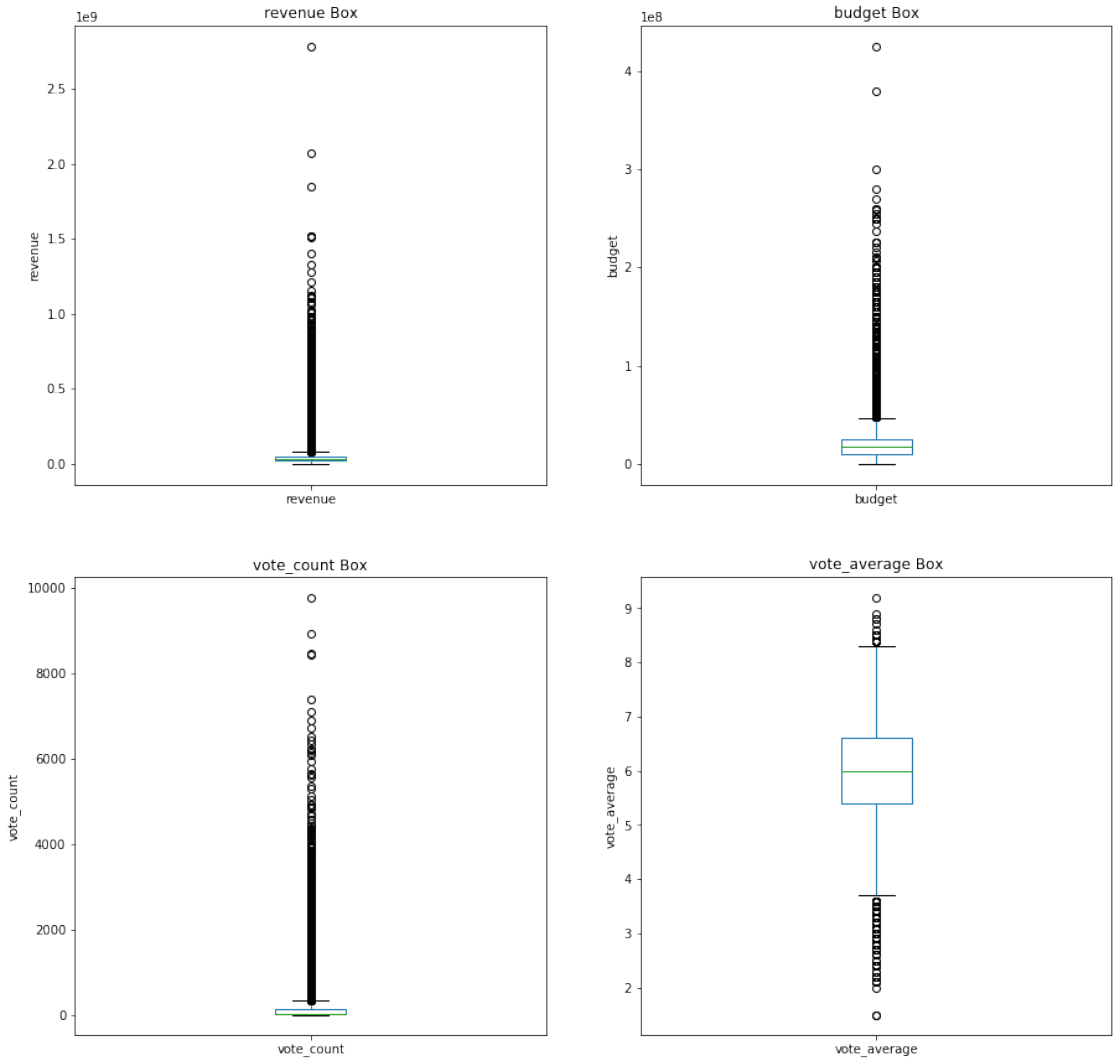
In [5]:
```
df_budget_train = df_all[df_all.budget != 0][['budget','release_year']]
df_revenue_train = df_all[df_all.revenue != 0][['revenue','release_year']]

gp_budget_median = df_budget_train.groupby('release_year')
gp_revenue_median = df_revenue_train.groupby('release_year')

def replace_zero_budget(zero):
    if zero.budget != 0:
        return zero.budget
    else:
        year = zero.release_year
        return gp_budget_median.get_group(year).budget.median()

def replace_zero_revenue(zero):
    if zero.revenue != 0:
        return zero.revenue
    else:
        year = zero.release_year
        return gp_revenue_median.get_group(year).revenue.median()


df_all['budget'] = df_all.apply(lambda x:replace_zero_budget(x),axis = 1)
df_all['revenue'] = df_all.apply(lambda x:replace_zero_revenue(x),axis = 1)
df_all.to_csv('movie_fixed.csv')
```

In [4]:
```
fig,axes=plt.subplots(2,2)

# 展示连续字段的基本统计描述信息
def show_describe_and_boxplot(column,axis):
    df_all[column].plot(figsize=(15,15),kind='box',title=column + ' Box',ax=axis)
    axis.set_ylabel(column)

    print(df_all[column].describe())

show_describe_and_boxplot('revenue',axes[0][0])
show_describe_and_boxplot('budget',axes[0][1])
show_describe_and_boxplot('vote_count',axes[1][0])
```

```
        show_describe_and_boxplot('vote_average',axes[1][1])

        plt.show()
```

```
count          10,866.00
mean       57,957,270.35
std       112,204,389.36
min                2.00
25%        21,676,985.50
50%        30,426,096.00
75%        46,321,980.00
max     2,781,505,847.00
Name: revenue, dtype: float64
count        10,866.00
mean     23,521,251.51
std      28,180,224.10
min              1.00
25%      10,000,000.00
50%      17,000,000.00
75%      25,000,000.00
max     425,000,000.00
Name: budget, dtype: float64
count   10,866.00
mean       217.39
std        575.62
min         10.00
25%         17.00
50%         38.00
75%        145.75
max      9,767.00
Name: vote_count, dtype: float64
count   10,866.00
mean         5.97
std          0.94
min          1.50
25%          5.40
50%          6.00
```

```
75%          6.60
max          9.20
Name: vote_average, dtype: float64
```



我们处理后的数据，仍然存在异常值，但数量已经非常少，整体数据的分布已经处于正常的情况，这些异常值在分析我们的问题时已不造成重大影响。

接下来我们对离散型的数据进行可用性检验。不过此时我们面临一个问题，genres 字段为“|”字符连接的多种类型，我们希望统计的是单个类型，因此需要先将 genres 进行处理。这里我们直接在原文件中使用 Excel 的“分列”功能将 genres 字段分隔为 genres1~genres5 五个字段，类型数量未达到 5 个的，对应字段填充为 Null。类似方式处理的还有 production_companies 和 keywords 字段，但它们的种类数量非常多，在不影响我们考察问题的前提下，将这两个字段的多余值进行合

并处理。全部的处理过程和处理后的数据如下：

```
In [13]: df_genres = pd.read_csv('movies_genres.csv')
         df_products = pd.read_csv('movies_products.csv')
         df_keywords = pd.read_csv('movies_keywords.csv')
         df_genres_melt = pd.melt(df_genres,id_vars=['id'],value_vars=['genres1','genres2','ge
         df_keywords_melt = pd.melt(df_keywords,id_vars=['id'],value_vars=['keywords1','keywor
         df_products_melt = pd.melt(df_products,id_vars=['id'],value_vars=['production_compani

         def combine_none_novel(keyword):
             if (keyword == 'based on novel') | (keyword == 'based on graphic novel'):
                 return 'based on novel'
             else:
                 return 'not based on novel'

         df_keywords_melt['value'] = df_keywords_melt.value.apply(lambda x:combine_none_novel(x

         def combine_producer(producer):
             if (producer != 'Universal Studios') & (producer != 'Paramount Pictures'):
                 return 'Others'
             else:
                 return producer

         df_products_melt['value'] = df_products_melt.value.apply(lambda x:combine_producer(x)

         print(df_genres_melt,df_keywords_melt,df_products_melt)
```

```
(           id variable            value
0      135397  genres1           Action
1       76341  genres1           Action
2      262500  genres1        Adventure
3      140607  genres1           Action
4      168259  genres1           Action
5      281957  genres1          Western
6       87101  genres1  Science Fiction
7      286217  genres1            Drama
8      211672  genres1           Family
9      150540  genres1           Comedy
```

10	206647	genres1	Action
11	76757	genres1	Science Fiction
12	264660	genres1	Drama
13	257344	genres1	Action
14	99861	genres1	Action
15	273248	genres1	Crime
16	260346	genres1	Crime
17	102899	genres1	Science Fiction
18	150689	genres1	Romance
19	131634	genres1	War
20	158852	genres1	Action
21	307081	genres1	Action
22	254128	genres1	Action
23	216015	genres1	Drama
24	318846	genres1	Comedy
25	177677	genres1	Action
26	214756	genres1	Comedy
27	207703	genres1	Crime
28	314365	genres1	Drama
29	294254	genres1	Action
...	...	...	...
53841	11244	genres5	Family
53846	22213	genres5	Thriller
53875	42700	genres5	Mystery
53882	16081	genres5	Music
53887	26039	genres5	Mystery
53910	5925	genres5	War
53914	751	genres5	Comedy
53959	8856	genres5	Family
54003	10017	genres5	Crime
54027	9930	genres5	Thriller
54049	47340	genres5	Thriller
54062	2362	genres5	Science Fiction
54081	17897	genres5	Fantasy
54115	11589	genres5	War
54120	11165	genres5	War
54147	5185	genres5	Romance

```
54168   21876  genres5             War
54187   29067  genres5         Foreign
54198   20391  genres5           Music
54207   42329  genres5         Western
54211   45522  genres5             War
54212    5227  genres5  Science Fiction
54225    6081  genres5          Family
54232   74849  genres5        TV Movie
54252   24961  genres5  Science Fiction
54291    2661  genres5           Crime
54299    5923  genres5         Romance
54307   29710  genres5          Action
54314   26268  genres5         Mystery
54315   15347  genres5         Foreign

[26960 rows x 3 columns],            id   variable               value
0      135397  keywords1  not based on novel
1       76341  keywords1  not based on novel
2      262500  keywords1      based on novel
3      140607  keywords1  not based on novel
4      168259  keywords1  not based on novel
5      281957  keywords1  not based on novel
6       87101  keywords1  not based on novel
7      286217  keywords1      based on novel
8      211672  keywords1  not based on novel
9      150540  keywords1  not based on novel
10     206647  keywords1  not based on novel
11      76757  keywords1  not based on novel
12     264660  keywords1  not based on novel
13     257344  keywords1  not based on novel
14      99861  keywords1  not based on novel
15     273248  keywords1  not based on novel
16     260346  keywords1  not based on novel
17     102899  keywords1  not based on novel
18     150689  keywords1  not based on novel
19     131634  keywords1  not based on novel
20     158852  keywords1  not based on novel
```

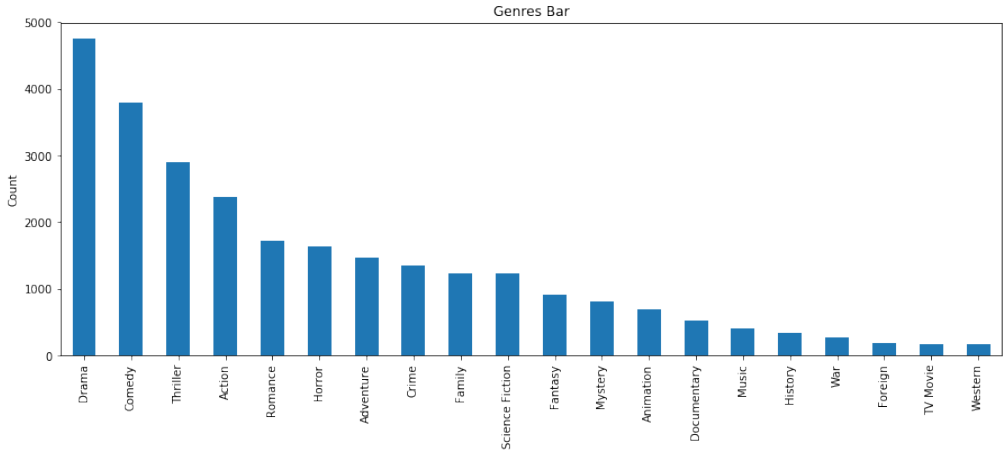
```
21     307081  keywords1  not based on novel
22     254128  keywords1  not based on novel
23     216015  keywords1      based on novel
24     318846  keywords1  not based on novel
25     177677  keywords1  not based on novel
26     214756  keywords1  not based on novel
27     207703  keywords1  not based on novel
28     314365  keywords1  not based on novel
29     294254  keywords1      based on novel
...       ...        ...                 ...
54263   17212  keywords5  not based on novel
54271   11623  keywords5  not based on novel
54272   40060  keywords5  not based on novel
54274    2182  keywords5  not based on novel
54276   31948  keywords5  not based on novel
54279   31657  keywords5  not based on novel
54284   13377  keywords5  not based on novel
54285    1714  keywords5  not based on novel
54286     396  keywords5  not based on novel
54287    3591  keywords5  not based on novel
54288    2525  keywords5  not based on novel
54289    1052  keywords5  not based on novel
54290     874  keywords5  not based on novel
54291    2661  keywords5  not based on novel
54293    6644  keywords5  not based on novel
54295    1888  keywords5  not based on novel
54297    3001  keywords5  not based on novel
54299    5923  keywords5  not based on novel
54302   22383  keywords5  not based on novel
54304   34388  keywords5  not based on novel
54305   42701  keywords5  not based on novel
54307   29710  keywords5  not based on novel
54310   17102  keywords5  not based on novel
54311   28763  keywords5  not based on novel
54312    2161  keywords5  not based on novel
54315   15347  keywords5  not based on novel
54318   31602  keywords5  not based on novel
```

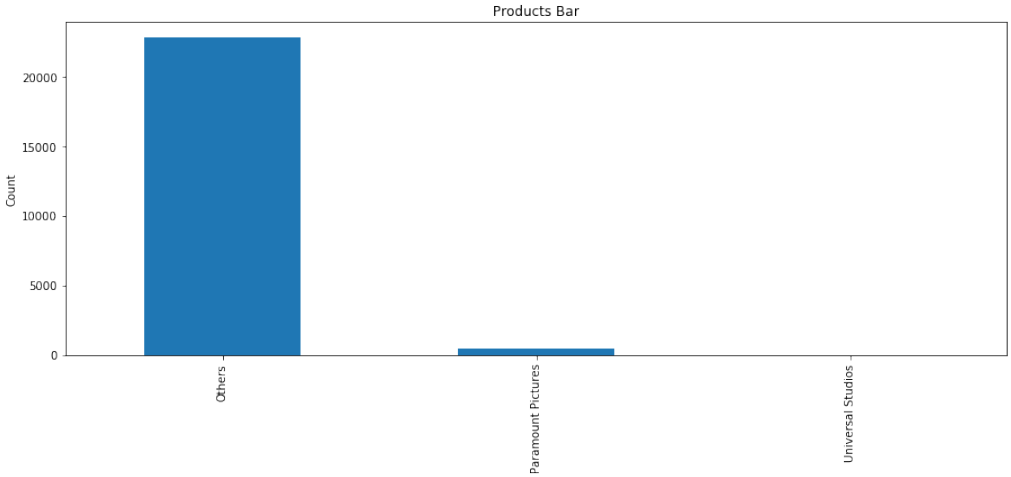
```
54319   13343  keywords5  not based on novel
54321    5921  keywords5  not based on novel
54329   22293  keywords5  not based on novel

[37450 rows x 3 columns],           id               variable                value
0      135397  production_companies1   Universal Studios
1       76341  production_companies1              Others
2      262500  production_companies1              Others
3      140607  production_companies1              Others
4      168259  production_companies1              Others
5      281957  production_companies1              Others
6       87101  production_companies1  Paramount Pictures
7      286217  production_companies1              Others
8      211672  production_companies1              Others
9      150540  production_companies1              Others
10     206647  production_companies1              Others
11      76757  production_companies1              Others
12     264660  production_companies1              Others
13     257344  production_companies1              Others
14      99861  production_companies1              Others
15     273248  production_companies1              Others
16     260346  production_companies1              Others
17     102899  production_companies1              Others
18     150689  production_companies1              Others
19     131634  production_companies1              Others
20     158852  production_companies1              Others
21     307081  production_companies1              Others
22     254128  production_companies1              Others
23     216015  production_companies1              Others
24     318846  production_companies1  Paramount Pictures
25     177677  production_companies1  Paramount Pictures
26     214756  production_companies1              Others
27     207703  production_companies1              Others
28     314365  production_companies1              Others
29     294254  production_companies1              Others
...       ...                    ...                 ...
52413   47886  production_companies5              Others
```

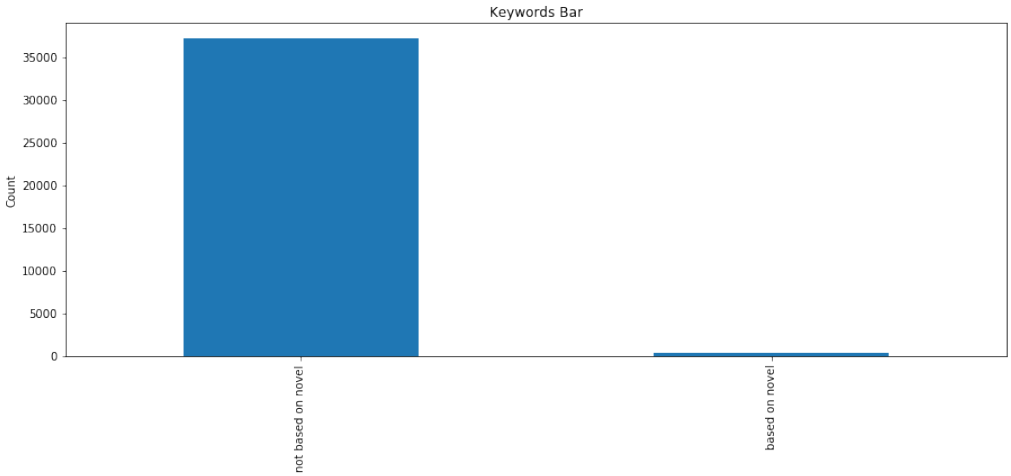
```
52438     857  production_companies5             Others
52461     929  production_companies5             Others
52462    6435  production_companies5             Others
52465    1878  production_companies5             Others
52471     795  production_companies5             Others
52491    9424  production_companies5             Others
52494     100  production_companies5             Others
52503    9945  production_companies5             Others
52523   11258  production_companies5             Others
52574    8129  production_companies5             Others
52599   37498  production_companies5             Others
52781     280  production_companies5             Others
52810   12627  production_companies5             Others
52812     339  production_companies5             Others
52853   20978  production_companies5             Others
53058     106  production_companies5             Others
53076     746  production_companies5             Others
53120   17238  production_companies5             Others
53128   39507  production_companies5             Others
53180   39929  production_companies5             Others
53239   16938  production_companies5             Others
53669    8446  production_companies5             Others
53685    9479  production_companies5             Others
53698    9350  production_companies5             Others
53711    9607  production_companies5             Others
53731    1634  production_companies5             Others
53771    2259  production_companies5             Others
53971   11416  production_companies5             Others
54033   13853  production_companies5             Others

[23227 rows x 3 columns])
```

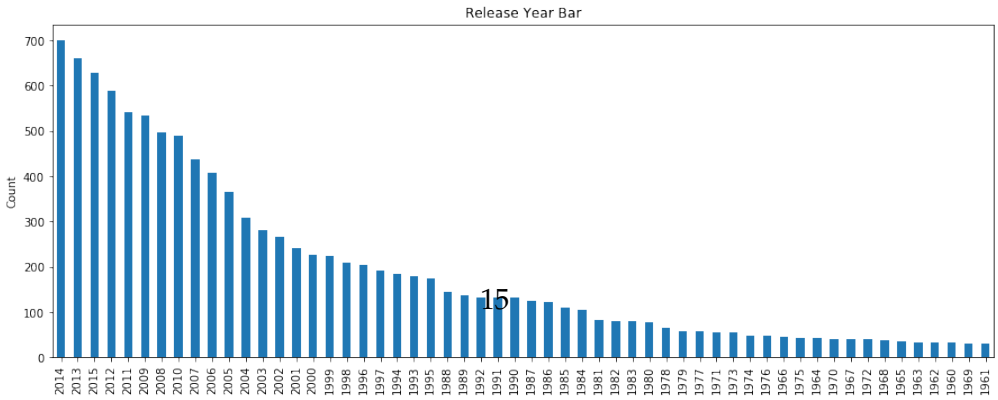
In [14]:
```
fig,axes=plt.subplots(4,1)

        plt.subplots_adjust(wspace=0.5,hspace=0.5);
```

```
def draw_bar(df,title,ylabel,ax_bar):
    ax_bar.set_ylabel(ylabel)
    df.value_counts().plot(kind='bar',ax=ax_bar,title=title,figsize=(15,30))

draw_bar(df_genres_melt['value'],'Genres Bar','Count',axes[0])
draw_bar(df_products_melt['value'],'Products Bar','Count',axes[1])
draw_bar(df_keywords_melt['value'],'Keywords Bar','Count',axes[2])
draw_bar(df_all.release_year,'Release Year Bar','Count',axes[3])

plt.show()
```


Genres Bar
Count
5000
4000
3000
2000
1000
0
Drama
Comedy
Thriller
Action
Romance
Horror
Adventure
Crime
Family
Science Fiction
Fantasy
Mystery
Animation
Documentary
Music
History
War
Foreign
TV Movie
Western


Products Bar
Count
20000
15000
10000
5000
0
Others
Paramount Pictures
Universal Studios


Keywords Bar
Count
35000
30000
25000
20000
15000
10000
5000
0
not based on novel
based on novel


Release Year Bar
Count
700
600
500
400
300
200
100
0
2014
2013
2015
2012
2011
2009
2008
2010
2007
2006
2005
2004
2003
2002
2001
2000
1999
1998
1996
1997
1994
1993
1995
1988
1989
1992
1991
1990
1987
1986
1985
1984
1981
1982
1983
1980
1978
1979
1977
1971
1973
1974
1976
1966
1975
1964
1970
1967
1972
1968
1965
1963
1962
1960
1969
1961

从上面的图形中可以看出，经过处理的四个离散字段，数据本身没有什么问题，能够满足我们考察问题使用，下面就进入使用 Tableau 进行数据探索的阶段。

1.1.2 数据探索

我们的数据探索，是根据提出的问题有目的地进行。

首先针对问题 1，随着时代发展电影类型的变化，我们用处理后的 genres 字段和 release_year 字段生成了如下图表：

https://public.tableau.com/shared/4DBW96DPS?:display_count=yes

从图 Q1_All 中可看出，Drama,Comedy 和 Thriller 类型的电影的上映数量一直处于比较稳定发展的状态，基本上一直占上映总数的前三位，其中 Drama 和 Thriller 的数量基本一直处于增长势头，Comedy 在和 Drama 短暂争夺过第一之后，近年来数量上保持稳定，比例上略有下降。处于第二梯队的是 Action,Horror,Adventure 和 Romance 类型的电影，其中 Horror 类型的电影增长幅度较大。最后，Western 和 War 类型的电影可以说衰退的非常厉害，跟各自的鼎盛时代相比，数量和比例近年来都处于很低的水平。当然，这仅仅是从上映数量上的观察，我们知道，近年来不乏有非常优秀的战争电影和西部电影上映，如果我们从票房或口碑的角度考察，可能会得出不同的结论。

针对问题 2，环球影业和派拉蒙两家电影公司各自出品的电影的数据差异，重点考察的是盈利水平的差异，我们使用处理后的数据得出如下图表：

https://public.tableau.com/views/genrescount_releaseyear/Q2Dashboard?:embed=y&:display_count=yes

从图中可以看出，两家制片商上映的影片总数分别为 U:522,P:431, 环球影视的电影数量略多于派拉蒙。两家在每年的票房总收入的较量上一直互有胜负，直到 2000 年之后，环球的总票房收入开始略胜一筹，2015 年的票房总收入更是大幅超过了派拉蒙。两家公司在电影类型上也是各有擅长，可以看到动作类型的电影是两家票房收入最多的一种，而且两家的动作电影票房收入也是非常接近，而在冒险和科幻类型电影上，派拉蒙的票房更高，环球则是在 Drama 和 Thriller 类型电影的票房上占据优势。还有值得注意的一点是，Animation 类型的电影虽然占比不大，但全部被派拉蒙垄断，可以推断如果双方在动画电影上发力将可能改变双方票房收入的格局。

针对问题 3，小说改编电影和非小说改编电影的表现有何不同，我们把考察重点放在了大众对影片的评价上。（这里发现原数据存在一个问题，有很多小说改编的电影没有被标记为改编，例如肖申克救赎、教父等，这里为分析方便仅以提供的数据为准。）为了讨论方便我们取了投票数量超过 500 的数据进行分析。图表如下所示：

https://public.tableau.com/views/genrescount_releaseyear/Q3Dashboard?:embed=y&:display_count=yes

首先根据散点图我们可以看出，非改编电影的绝对数量上远远超过改编电影，且分布趋于分散，大部分集中在投票数 2000 以下，得分 5.5~7.5 之间的区域。口碑特别好（投票数多，平均分高）的电影和特别不好（投票数少，得分低）的都比较突出，最低得分只有 3.9 分；而改编电影的分布就相对集中，可以说相较非改编电影而言，表现比较 “稳定”。但两者总体的分布形状还是比较接近

的。从各自平均得分 Top 10 的电影图表中也能看出，对两种类型的电影中的佼佼者来说，在口碑方面的差别并不显著。值得注意的是在 1994 年诞生了两部在各自类别中都处于得分首位的电影，可以称得上是影迷们最开心的一年了。

针对问题 4，我制作了一个 StoryBoard，尝试分析电影高票房与其成本的关系，链接如下：

Dashboard: https://public.tableau.com/shared/TNMMJJN7Q?:display_count=yes Story: https://public.tableau.com/shared/NX83STMBZ?:display_count=yes