Movie-Domestic-Gross

Import the packages needed to perform the analysis

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [2]:

```
# Import the data
mov = pd.read_csv('Movie-Domestic-Gross.csv', encoding = 'latin1')
```

In [3]:

```
# Explore the dataset
mov.head()
```

Out[3]:

	Day of Week	Director	Genre	Movie Title	Release Date	Studio	Adjusted Gross ($mill)	Budget ($mill)	Gro ($m
0	Friday	Brad Bird	action	Tomorrowland	22/05/2015	Buena Vista Studios	202.1	170.0	202
1	Friday	Scott Waugh	action	Need for Speed	14/03/2014	Buena Vista Studios	204.2	66.0	203
2	Friday	Patrick Hughes	action	The Expendables 3	15/08/2014	Lionsgate	207.1	100.0	206
3	Friday	Phil Lord, Chris Miller	comedy	21 Jump Street	16/03/2012	Sony	208.8	42.0	201
4	Friday	Roland Emmerich	action	White House Down	28/06/2013	Sony	209.7	150.0	205

In [4]:

```
# Check the summary of the dataframe
mov.describe()
```

Out[4]:

	Budget ($mill)	IMDb Rating	MovieLens Rating	Overseas%	Profit%	Runtime (min)	US (
count	608.000000	608.000000	608.000000	608.000000	608.000000	608.000000	608.
mean	92.467928	6.923849	3.340378	57.698849	719.278783	117.781250	167.
std	59.421407	0.925890	0.454071	12.334237	1942.807248	23.179122	92.4
min	0.600000	3.600000	1.490000	17.200000	7.700000	30.000000	0.00
25%	45.000000	6.375000	3.037500	49.900000	201.850000	100.000000	106.
50%	80.000000	6.900000	3.365000	58.200000	338.550000	116.000000	141.
75%	130.000000	7.600000	3.672500	66.300000	650.100000	130.250000	202.
max	300.000000	9.200000	4.500000	100.000000	41333.300000	238.000000	760.

In [5]:

```
# Check the structure of the dataframe
mov.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 608 entries, 0 to 607
Data columns (total 18 columns):
Day of Week               608 non-null object
Director                  608 non-null object
Genre                     608 non-null object
Movie Title               608 non-null object
Release Date              608 non-null object
Studio                    608 non-null object
Adjusted Gross ($mill)    608 non-null object
Budget ($mill)            608 non-null float64
Gross ($mill)             608 non-null object
IMDb Rating               608 non-null float64
MovieLens Rating          608 non-null float64
Overseas ($mill)          608 non-null object
Overseas%                 608 non-null float64
Profit ($mill)            608 non-null object
Profit%                   608 non-null float64
Runtime (min)             608 non-null int64
US ($mill)                608 non-null float64
Gross % US                608 non-null float64
dtypes: float64(7), int64(1), object(10)
memory usage: 61.8+ KB
```

In [6]:

```
# Explore the categorical variable Studio
print(mov.Studio.unique())
print(len(mov.Studio.unique()))
```

```
['Buena Vista Studios' 'Lionsgate' 'Sony' 'Universal' 'Paramount Pictures'
 'WB' 'Weinstein Company' 'UA Entertainment' 'WB/New Line'
 'New Line Cinema' 'Fox' 'TriStar' 'Relativity Media' 'Screen Gems'
 'StudioCanal' 'Fox Searchlight Pictures' 'MiraMax' 'Path_ Distribution'
 'DreamWorks' 'Lionsgate Films' 'Revolution Studios' 'Dimension Films'
 'USA' 'Lionsgate/Summit' 'Sony Picture Classics' 'Pacific Data/DreamWorks'
 'Disney' 'Art House Studios' 'Colombia Pictures' 'Gramercy Pictures'
 'Summit Entertainment' 'Vestron Pictures' 'MGM' 'Orion' 'IFC'
 'New Market Films']
36
```

In [7]:

```
# Explore the categorical variable Genre
print(mov.Genre.unique())
print(len(mov.Genre.unique()))
```

```
['action' 'comedy' 'adventure' 'horror' 'animation' 'biography' 'drama'
 'musical' 'sci-fi' 'crime' 'romance' 'fantasy' 'mystery' 'thriller'
 'documentary']
15
```

In [8]:

```
# Filter the dataframe by genre
mov2 = mov[(mov.Genre == 'action') | (mov.Genre == 'adventure') | (mov.Genre == 'animation'
           (mov.Genre == 'comedy') | (mov.Genre == 'drama')]
mov2.Genre.unique()
#len( mov2[mov2.Genre == 'action'])
#len( mov2[mov2.Genre == 'adventure'])
#len( mov2[mov2.Genre == 'animation'])
#len( mov2[mov2.Genre == 'comedy'])
#len( mov2[mov2.Genre == 'drama'])
```

Out[8]:

```
array(['action', 'comedy', 'adventure', 'animation', 'drama'], dtype=object)
```

In [9]:

```
# Filter the dataframe by studio
mov3 = mov2[(mov2.Studio == 'Buena Vista Studios') | (mov2.Studio == 'Fox') | (mov2.Studio
            (mov2.Studio == 'Sony') | (mov2.Studio == 'Universal') | (mov2.Studio == 'WB')]
mov3.Studio.unique()
```

Out[9]:

```
array(['Buena Vista Studios', 'Sony', 'Universal', 'WB',
       'Paramount Pictures', 'Fox'], dtype=object)
```

In [10]:

```
# Check how the filters worked
print (mov3.Genre.unique())
print (mov3.Studio.unique())
print (len(mov3))
```

```
['action' 'comedy' 'adventure' 'animation' 'drama']
['Buena Vista Studios' 'Sony' 'Universal' 'WB' 'Paramount Pictures' 'Fox']
423
```

In [11]:

```
mov3.columns=['DayofWeek', 'Director', 'Genre', 'MovieTitle', 'ReleaseDate',
       'Studio', 'AdjustedGross$mill', 'BudgetDollarmill', 'GrossMillions',
       'IMDbRating', 'MovieLensRating', 'Overseas$mill', 'Overseas%',
       'Profit$mill', 'Profit%', 'RuntimeMin', 'US$mill',
       'Gross%US']
```

In [12]:

```
mov3.Genre  = mov3.Genre.astype('category')
```

In [13]:

```
mov3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 423 entries, 0 to 606
Data columns (total 18 columns):
DayofWeek             423 non-null object
Director              423 non-null object
Genre                 423 non-null category
MovieTitle            423 non-null object
ReleaseDate           423 non-null object
Studio                423 non-null object
AdjustedGross$mill    423 non-null object
BudgetDollarmill      423 non-null float64
GrossMillions         423 non-null object
IMDbRating            423 non-null float64
MovieLensRating       423 non-null float64
Overseas$mill         423 non-null object
Overseas%             423 non-null float64
Profit$mill           423 non-null object
Profit%               423 non-null float64
RuntimeMin            423 non-null int64
US$mill               423 non-null float64
Gross%US              423 non-null float64
dtypes: category(1), float64(7), int64(1), object(9)
memory usage: 45.0+ KB
```

In [14]:

```
mov3.columns
```

Out[14]:

```
Index(['DayofWeek', 'Director', 'Genre', 'MovieTitle', 'ReleaseDate', 'Studi
o',
       'AdjustedGross$mill', 'BudgetDollarmill', 'GrossMillions', 'IMDbRatin
g',
       'MovieLensRating', 'Overseas$mill', 'Overseas%', 'Profit$mill',
       'Profit%', 'RuntimeMin', 'US$mill', 'Gross%US'],
      dtype='object')
```

In [15]:

```
# Define the style
sns.set(style="darkgrid", palette="muted", color_codes=True)
# Plot the boxsplots

ax = sns.boxplot(data=mov3, x='Genre', y='Gross%US', orient='v',color='lightgray',showflier
plt.setp(ax.artists, alpha=0.5)

# Add in points to show each observation
sns.stripplot(x='Genre', y='Gross%US', data=mov3, jitter=True, size=6, linewidth=0, hue = '

ax.axes.set_title('Domestic Gross % by Genre',fontsize=30)
ax.set_xlabel('Genre',fontsize=20)
ax.set_ylabel('Gross % US',fontsize=20)

# Define where to place the legend
ax.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

Out[15]:

```
<matplotlib.legend.Legend at 0x86fc3f0>
```