Loading Libraries and Dataset

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprec
ated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

In [2]:
```
dfx = pd.read_excel('/Movie300.xlsx')
dfx.head()
```

Out[2]:

	Movie_name	Movie_Genre	Movie_Certification	Release	Runtime	Lead_Actor	Lead_Actress	Movie_Critic_Rating	Movie_User_Rating	Movie_S
0	Kannum Kannum Kollaiyadithaal	Thriller	U	28 Feb 2020	2 hrs 2 mins	Dulquer Salmaan	Ritu Varma	3.5	4.3	Pe displays
1	Oh My Kadavule	Comedy	UA	14 Feb 2020	2 hrs 31 mins	Ashok Selvan	Ritika Singh	3.5	3.4	Desp cor turr
2	Psycho	Thriller	A	24 Jan 2020	2 hrs 14 mins	Udhayanidhi Stalin	Aditi Rao Hydari	3.5	3.3	fi comb per
3	Dharala Prabhu	Comedy	UA	13 Mar 2020	2 hrs 2 mins	Harish Kalyan	Tanya Hope	3.0	3.3	ba score ef
4	Gypsy	Drama	UA	06 Mar 2020	2 hrs 25 mins	Jiiva	Natasha Singh	3.0	3.2	Gyp talks politicis

Dataset Cleaning and some preliminary steps

In [3]:
```
dfx['Release_Month'] = dfx['Release'].apply(lambda x: x.split(' ')[1])
dfx['Release_Month'].head()
```

Out[3]:
```
0    Feb
1    Feb
2    Jan
3    Mar
4    Mar
Name: Release_Month, dtype: object
```

In [4]:
```
dfx.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Movie_name           300 non-null    object 
 1   Movie_Genre          300 non-null    object 
 2   Movie_Certification  300 non-null    object 
 3   Release              300 non-null    object 
 4   Runtime              300 non-null    object 
 5   Lead_Actor           235 non-null    object 
 6   Lead_Actress         230 non-null    object 
 7   Movie_Critic_Rating  300 non-null    float64
 8   Movie_User_Rating    300 non-null    float64
 9   Movie_Synopsis       297 non-null    object 
 10  Movie_Full_Cast      299 non-null    object 
 11  Release_Month        300 non-null    object 
dtypes: float64(2), object(10)
memory usage: 28.2+ KB
```

In [5]:
```
dfx.fillna
```

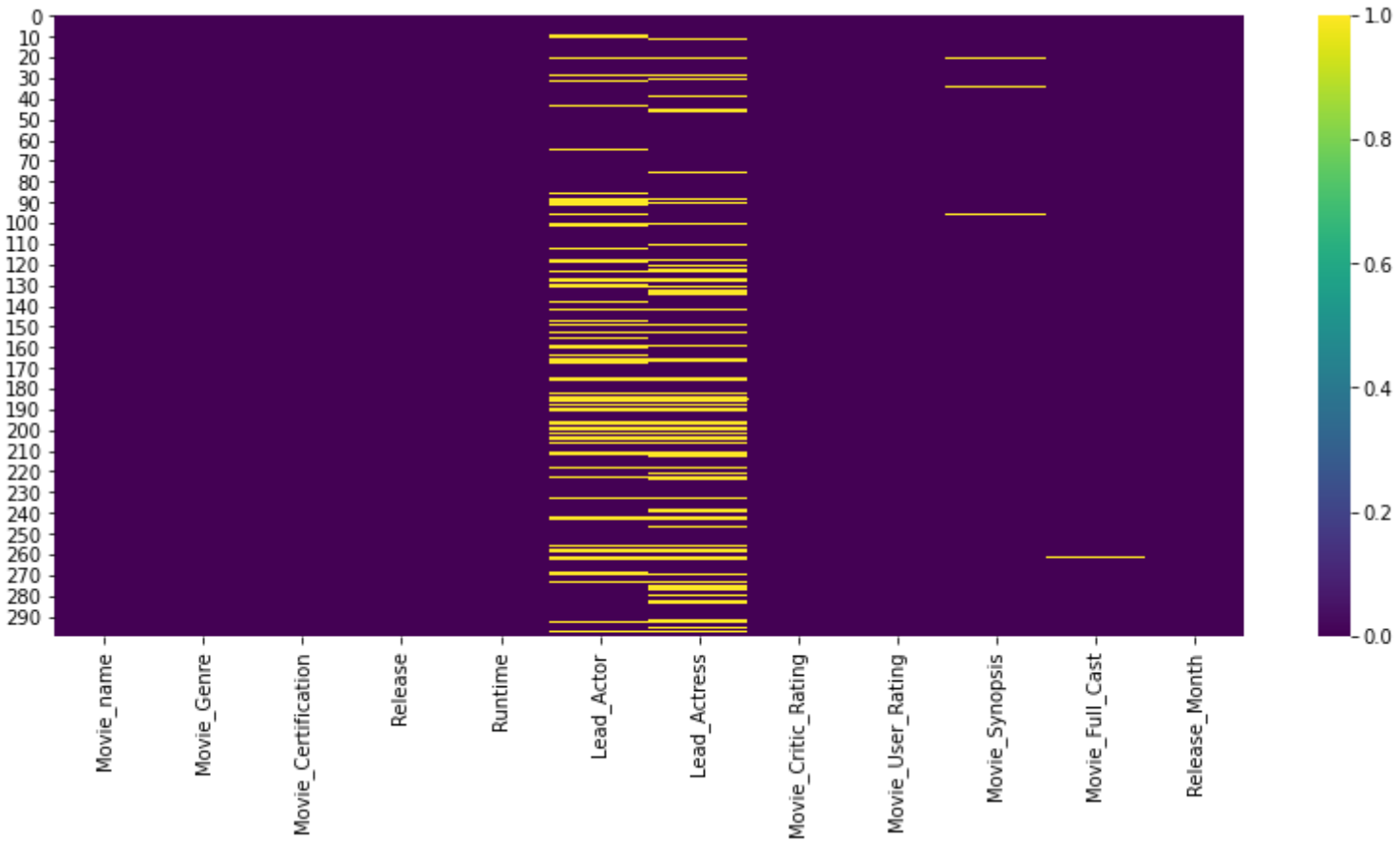
Out[5]:
```
<bound method DataFrame.fillna of                         Movie_name  ... Release_Month
0    Kannum Kannum Kollaiyadithaal  ...           Feb
1                  Oh My Kadavule  ...           Feb
2                          Psycho  ...           Jan
3                  Dharala Prabhu  ...           Mar
4                           Gypsy  ...           Mar
..                            ...  ...           ...
295                   Nootrenbadhu  ...           Jun
296                 Ponnar Shankar  ...           Apr
297              Nadunisi Naaygal  ...           Feb
298                      Ilaignan  ...           Jan
299                     Mappillai  ...           Apr

[300 rows x 12 columns]>
```

In [6]:
```
dfx.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Movie_name           300 non-null    object 
 1   Movie_Genre          300 non-null    object 
 2   Movie_Certification  300 non-null    object 
 3   Release              300 non-null    object 
 4   Runtime              300 non-null    object 
 5   Lead_Actor           235 non-null    object 
 6   Lead_Actress         230 non-null    object 
 7   Movie_Critic_Rating  300 non-null    float64
 8   Movie_User_Rating    300 non-null    float64
 9   Movie_Synopsis       297 non-null    object 
 10  Movie_Full_Cast      299 non-null    object 
 11  Release_Month        300 non-null    object 
dtypes: float64(2), object(10)
memory usage: 28.2+ KB
```

In [7]:
```
plt.figure(figsize=(14,6))
sb.heatmap(dfx.isnull(), cmap="viridis")
```

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f53b6c6a2e8>



In [8]:
```
dfx['Lead_Actor'].isnull().sum()
```

Out[8]: 65

In [9]:
```
dfx['Lead_Actress'].isnull().sum()
```

Out[9]: 70

In [10]:
```
dfy = dfx.copy()
```

In [11]:
```
dfy.head()
```

Out[11]:

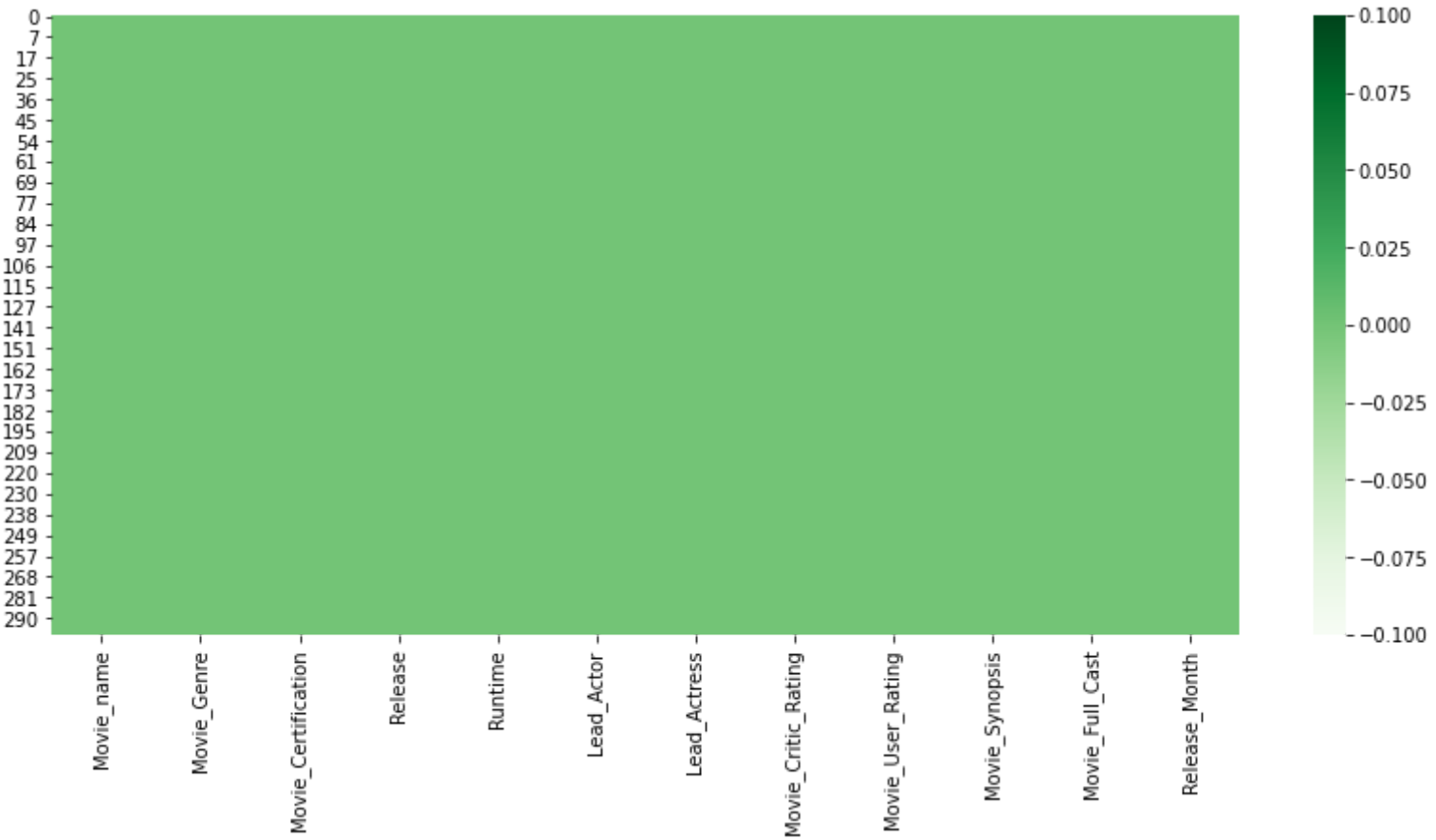
	Movie_name	Movie_Genre	Movie_Certification	Release	Runtime	Lead_Actor	Lead_Actress	Movie_Critic_Rating	Movie_User_Rating	Movie_S
0	Kannum Kannum Kollaiyadithaal	Thriller	U	28 Feb 2020	2 hrs 2 mins	Dulquer Salmaan	Ritu Varma	3.5	4.3	Pe displays
1	Oh My Kadavule	Comedy	UA	14 Feb 2020	2 hrs 31 mins	Ashok Selvan	Ritika Singh	3.5	3.4	Desp cor turr
2	Psycho	Thriller	A	24 Jan 2020	2 hrs 14 mins	Udhayanidhi Stalin	Aditi Rao Hydari	3.5	3.3	fi comb per
3	Dharala Prabhu	Comedy	UA	13 Mar 2020	2 hrs 2 mins	Harish Kalyan	Tanya Hope	3.0	3.3	ba score ef
4	Gypsy	Drama	UA	06 Mar 2020	2 hrs 25 mins	Jiiva	Natasha Singh	3.0	3.2	Gyp talks politicisa

In [12]:
```
dfy.dropna(inplace=True)
```

In [13]:

```
plt.figure(figsize=(14,6))
sb.heatmap(dfy.isnull(), cmap="Greens")

#'Accent', 'Accent_r', 'Blues', 'Blues_r', 'BrBG', 'BrBG_r', 'BuGn', 'BuGn_r', 'BuPu', 'BuPu_r', 'CMRmap',
#'CMRmap_r', 'Dark2', 'Dark2_r', 'GnBu', 'GnBu_r', 'Greens', 'Greens_r', 'Greys', 'Greys_r', 'OrRd', 'OrRd_r',
#'Oranges', 'Oranges_r', 'PRGn', 'PRGn_r', 'Paired', 'Paired_r', 'Pastel1', 'Pastel1_r', 'Pastel2', 'Pastel2_r',
#'PiYG', 'PiYG_r', 'PuBu', 'PuBuGn', 'PuBuGn_r', 'PuBu_r', 'PuOr', 'PuOr_r', 'PuRd', 'PuRd_r', 'Purples', 'Purples_r',
#'RdBu', 'RdBu_r', 'RdGy', 'RdGy_r', 'RdPu', 'RdPu_r', 'RdYlBu', 'RdYlBu_r', 'RdYlGn', 'RdYlGn_r', 'Reds', 'Reds_r',
#'Set1', 'Set1_r', 'Set2', 'Set2_r', 'Set3', 'Set3_r', 'Spectral', 'Spectral_r', 'Wistia', 'Wistia_r', 'YlGn', 'YlGnB
u',
#'YlGnBu_r', 'YlGn_r', 'YlOrBr', 'YlOrBr_r', 'YlOrRd', 'YlOrRd_r', 'afmhot', 'afmhot_r', 'autumn', 'autumn_r', 'binar
y',
#'binary_r', 'bone', 'bone_r', 'brg', 'brg_r', 'bwr', 'bwr_r', 'cividis', 'cividis_r', 'cool', 'cool_r', 'coolwarm',
#'coolwarm_r', 'copper', 'copper_r', 'cubehelix', 'cubehelix_r', 'flag', 'flag_r', 'gist_earth', 'gist_earth_r',
#'gist_gray', 'gist_gray_r', 'gist_heat', 'gist_heat_r', 'gist_ncar', 'gist_ncar_r', 'gist_rainbow', 'gist_rainbow_r',
#'gist_stern', 'gist_stern_r', 'gist_yarg', 'gist_yarg_r', 'gnuplot', 'gnuplot2', 'gnuplot2_r', 'gnuplot_r', 'gray',
#'gray_r', 'hot', 'hot_r', 'hsv', 'hsv_r', 'icefire', 'icefire_r', 'inferno', 'inferno_r', 'jet', 'jet_r', 'magma',
#'magma_r', 'mako', 'mako_r', 'n...
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7f53b30a8048>



In [14]:
```
dfy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 209 entries, 0 to 299
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Movie_name           209 non-null    object 
 1   Movie_Genre          209 non-null    object 
 2   Movie_Certification  209 non-null    object 
 3   Release              209 non-null    object 
 4   Runtime              209 non-null    object 
 5   Lead_Actor           209 non-null    object 
 6   Lead_Actress         209 non-null    object 
 7   Movie_Critic_Rating  209 non-null    float64
 8   Movie_User_Rating    209 non-null    float64
 9   Movie_Synopsis       209 non-null    object 
 10  Movie_Full_Cast      209 non-null    object 
 11  Release_Month        209 non-null    object 
dtypes: float64(2), object(10)
memory usage: 21.2+ KB
```

Genre Analysis

In [15]:
```
dfx['Movie_Genre'].unique()
```

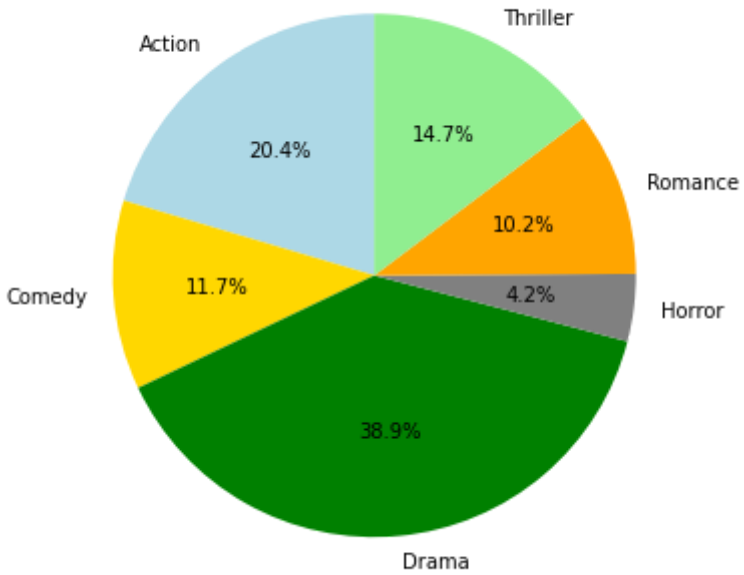
Out[15]:
```
array(['Thriller', 'Comedy', 'Drama', 'Action', 'Family', 'Crime',
       'Adventure', 'Musical', 'Biography', 'Sports', 'Romance', 'Sci-Fi',
       'Mystery', 'Horror', 'Documentary', 'Short', 'History', 'Fantasy'],
      dtype=object)
```

In [16]:
```
genre = dfx.groupby('Movie_Genre')['Movie_Genre'].count()
genre
```

Out[16]:
```
Movie_Genre
Action          54
Adventure        2
Biography        2
Comedy          31
Crime           10
Documentary      1
Drama          103
Family           1
Fantasy          2
History          3
Horror          11
Musical          2
Mystery          3
Romance         27
Sci-Fi           4
Short            1
Sports           4
Thriller        39
Name: Movie_Genre, dtype: int64
```

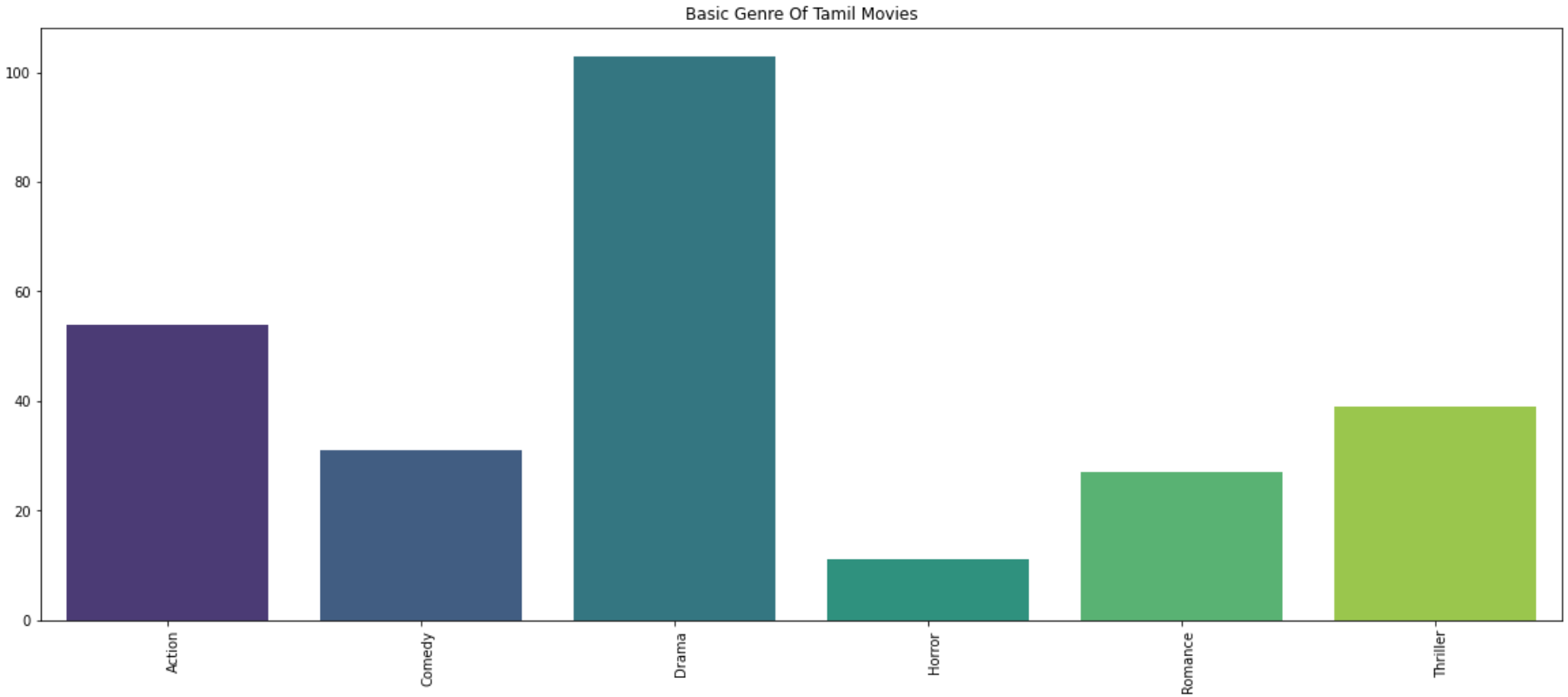
Pie Chart Representation of basic genre

In [17]:
```
genre_general = ['Action', 'Comedy', 'Drama', 'Horror', 'Romance', 'Thriller']
genre_general_values = [54, 31, 103, 11, 27, 39]

#pie chart
colors = ['lightblue', 'gold', 'green','grey', 'orange', 'lightgreen' ]
plt.subplots(figsize=(14,6))
plt.pie(genre_general_values,labels=genre_general, colors=colors, startangle = 90, autopct='%.1f%%')
plt.show()
```



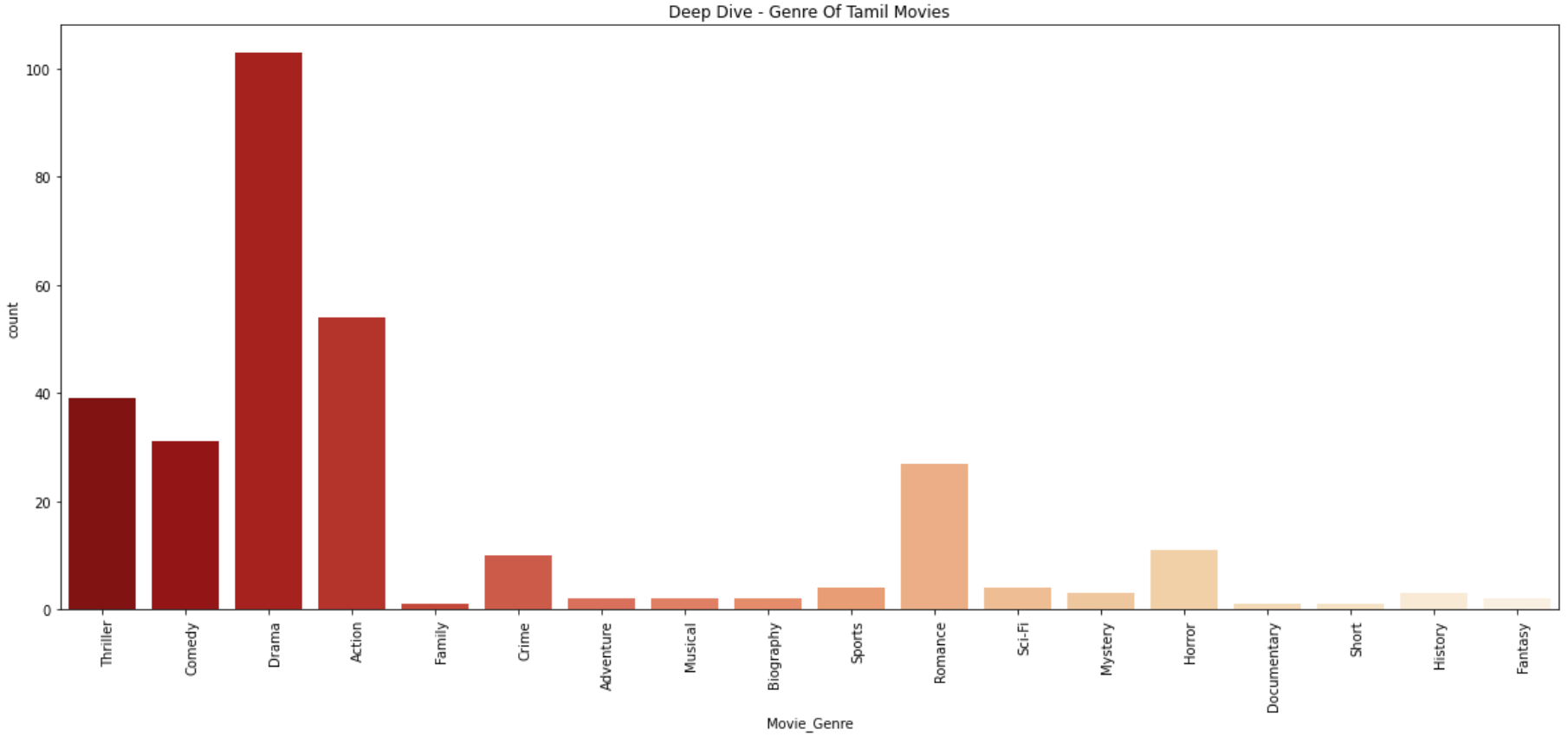
Bar Chart Representation of basic genre

In [18]:
```
plt.figure(figsize=(20,8))
x = genre_general
y = genre_general_values
chart1 = sb.barplot(x, y, palette='viridis')
chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
chart1.set_title('Basic Genre Of Tamil Movies')
plt.show()
```



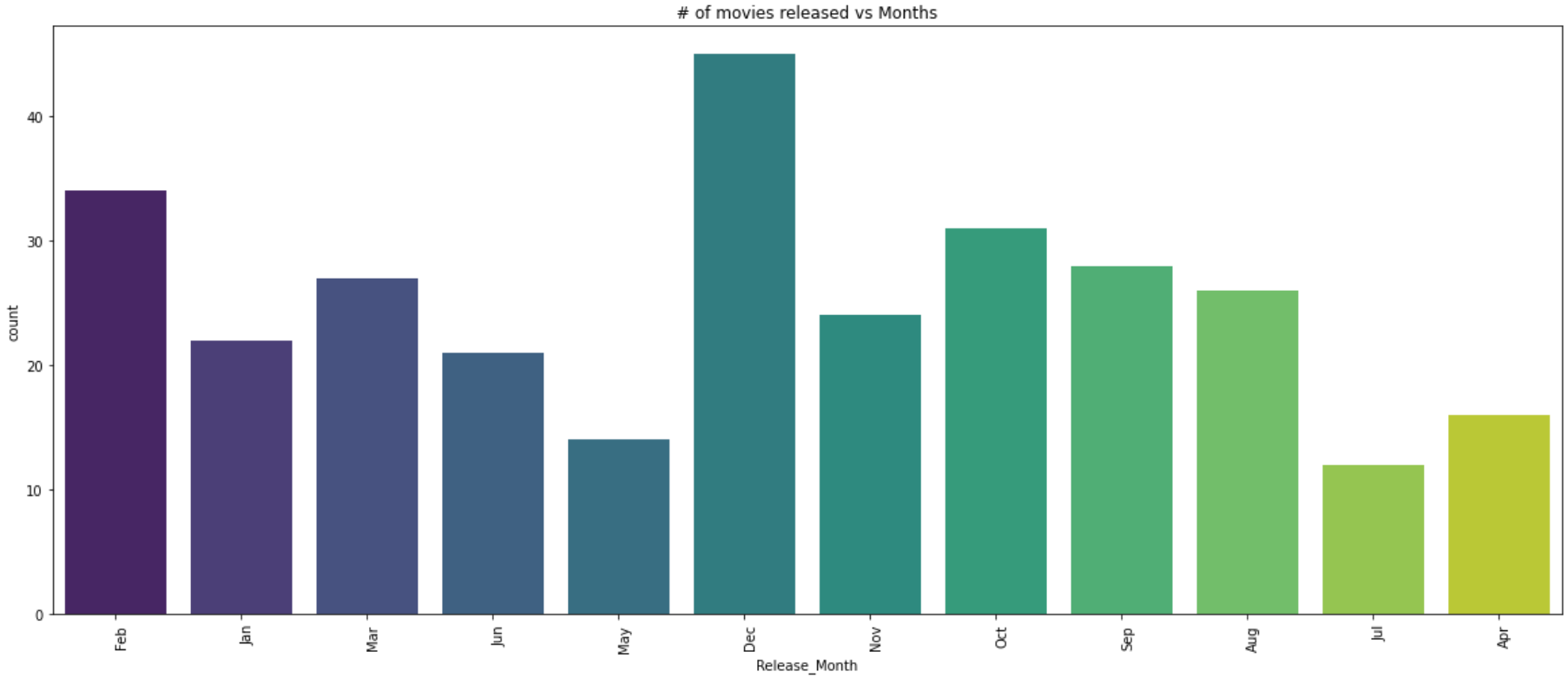
Depth Analysis of Genre

In [19]:
```
plt.figure(figsize=(20,8))
x = dfx['Movie_Genre']
chart1 = sb.countplot(x, data=dfx, palette='OrRd_r')
chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
chart1.set_title('Deep Dive - Genre Of Tamil Movies')
plt.show()
```



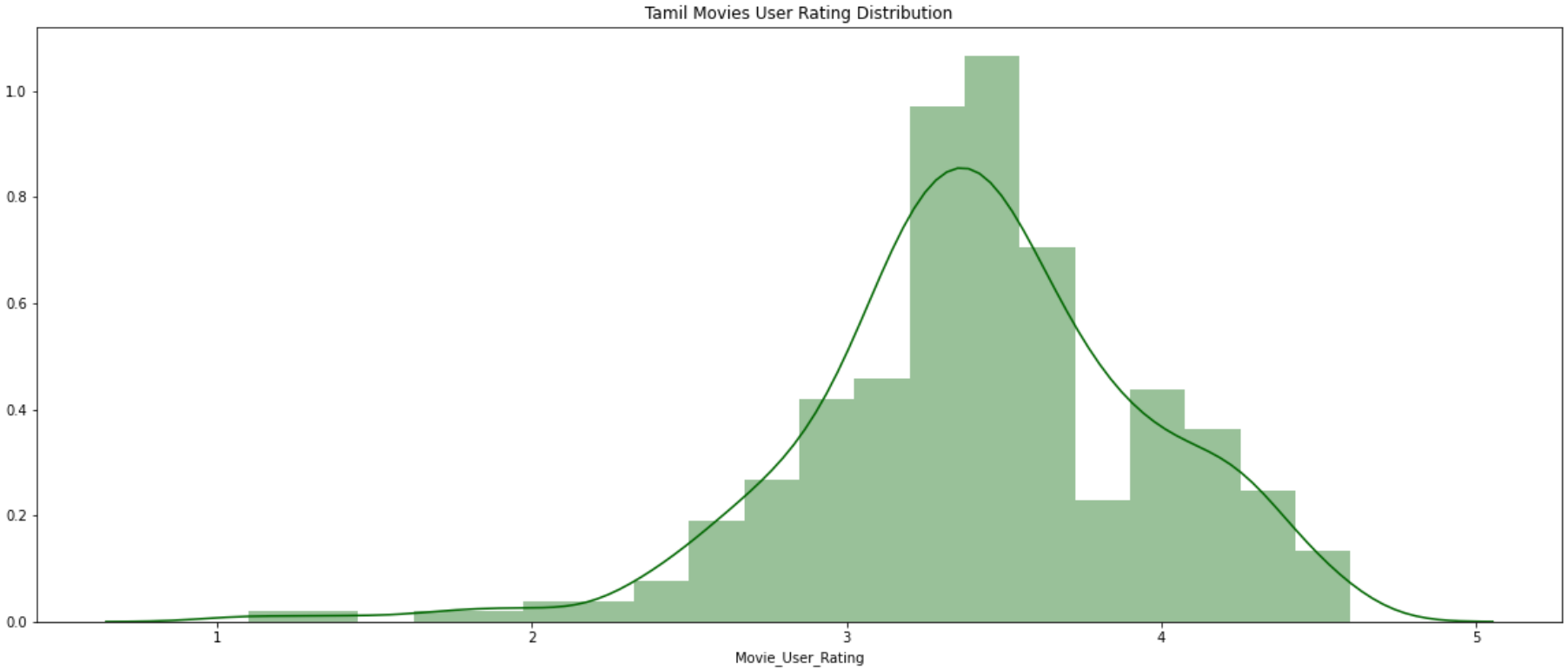
Month-wise Visualization of Movie releases

In [20]:
```
plt.figure(figsize=(20,8))
chart1 = sb.countplot(x=dfx['Release_Month'], data=dfx, palette='viridis')
chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
chart1.set_title('# of movies released vs Months')
plt.show()
```



User Rating Distribution Plot

In [21]:
```
plt.figure(figsize=(20,8))
chart2 = sb.distplot(dfx['Movie_User_Rating'],  color="#006600")
chart2.set_title('Tamil Movies User Rating Distribution')
plt.show()
```

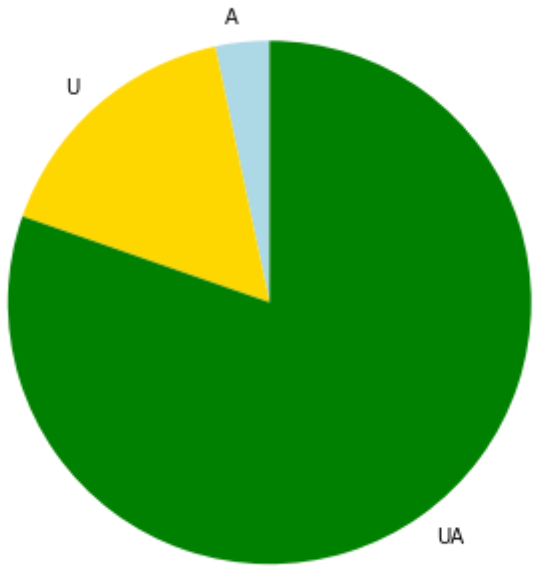


Movie Certifications Analysis

Pie Chart Representation of Movie Censorboard Certification

In [22]:
```
mc = dfx.groupby('Movie_Certification')['Movie_Certification'].count()
mc_entities = ['A', 'U', 'UA']
mc_values = [10,49,241]
```

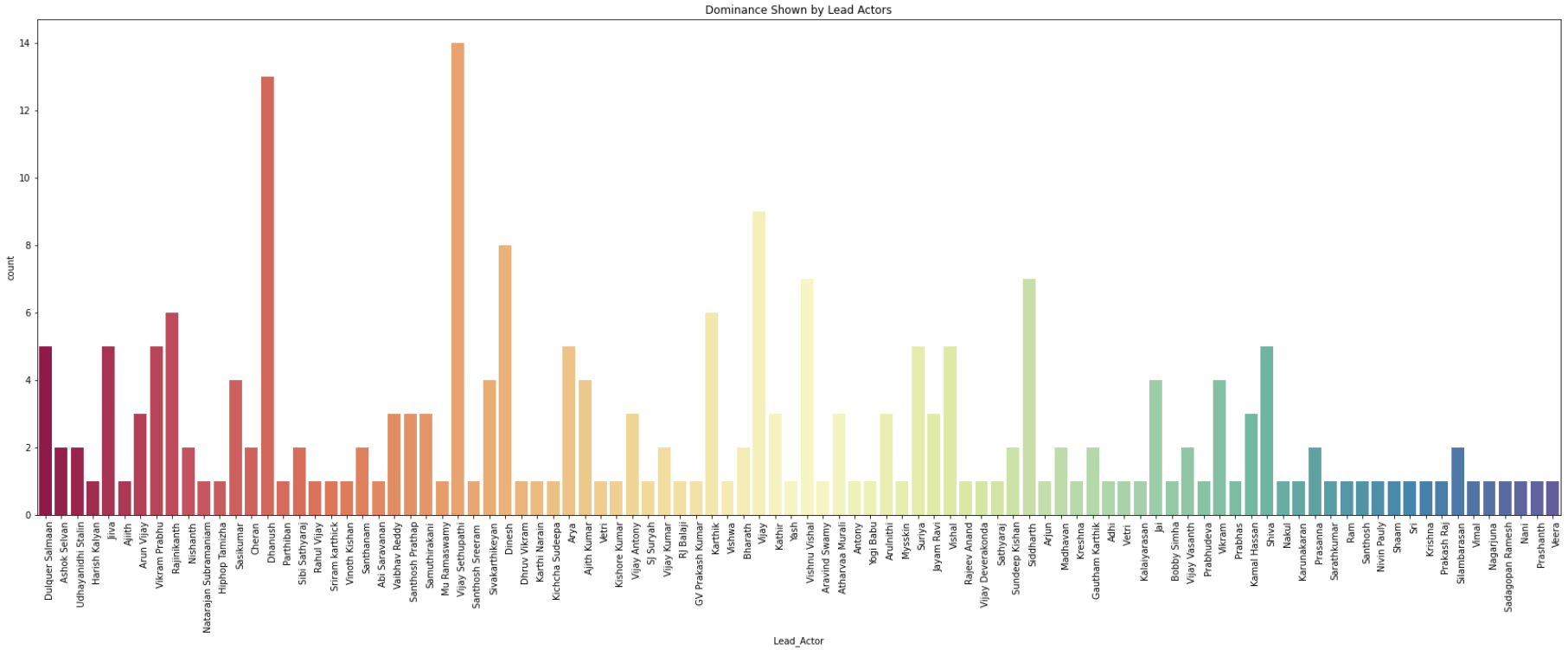
In [23]:
```
plt.subplots(figsize=(14,6))
colors = ['lightblue', 'gold', 'green']
plt.pie(mc_values, labels = mc_entities, colors=colors, startangle = 90)
plt.show()
```



Analysis of Lead Actors and Lead Actresses of the Decade

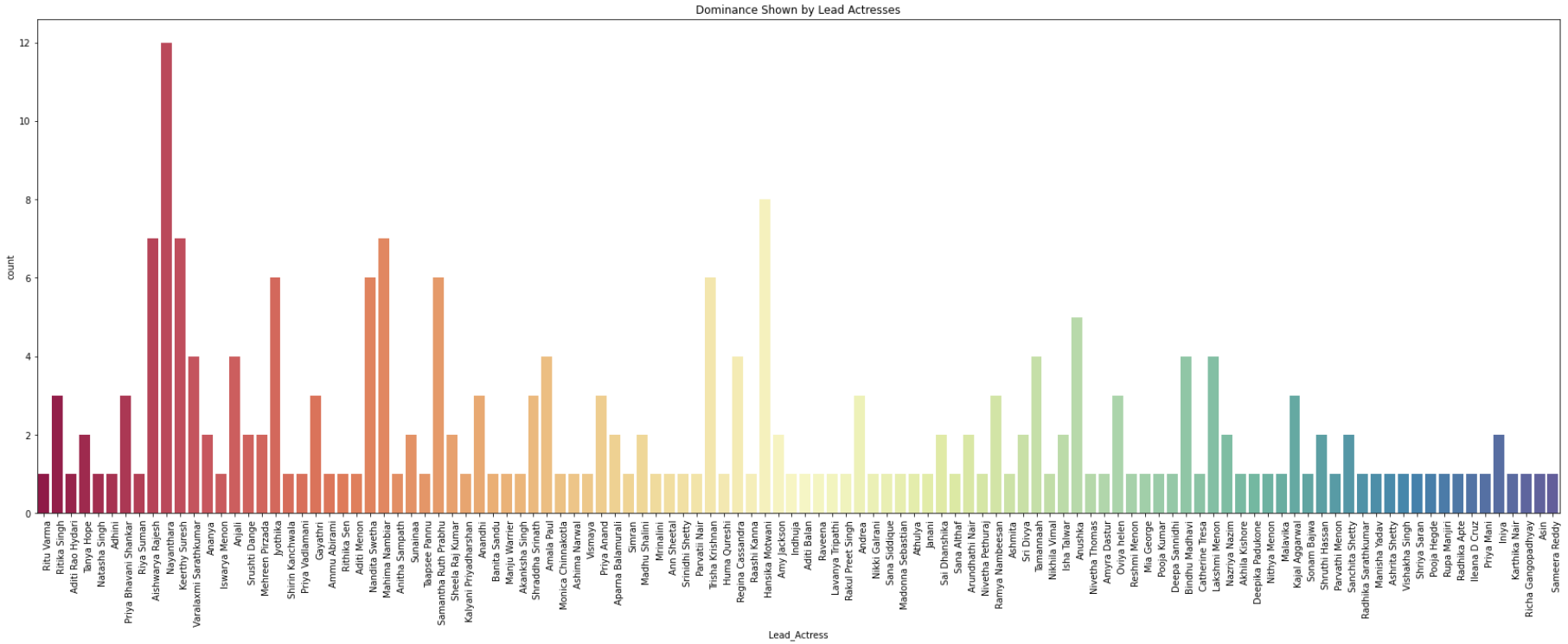
Dominance comparison of Lead Actors

In [24]:
```
plt.figure(figsize=(30,10))
chart1 = sb.countplot(x=dfx['Lead_Actor'], data=dfx, palette='Spectral')
chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
chart1.set_title('Dominance Shown by Lead Actors')
plt.show()
```



Dominance comparison of Lead Actresses

In [25]:
```
plt.figure(figsize=(30,10))
chart1 = sb.countplot(x=dfx['Lead_Actress'], data=dfx, palette='Spectral')
chart1.set_xticklabels(chart1.get_xticklabels(), rotation=90, horizontalalignment='left')
chart1.set_title('Dominance Shown by Lead Actresses')
plt.show()
```



Conclusion

Presenting the Rockstars of Kollywood in the decade 2011-2020