CSC6730 Project2

Ruixie Fang

2.Load HollywoodsMostProfitableStories.csv and use Plotly to create the following charts. Every figure must include a title. Each axis must be labelled.

In [2]:
```
import os
os.chdir("D:\GSU\Study\Summer\Data_Vis\Proj2\Proectj2")
```

In [3]:
```
import pandas as pd
hp = pd.read_csv("HollywoodsMostProfitableStories.csv")
```

2.a.A bar chart showing the profitability of the film. The X axis is the film. The Y axis is the profitability. The bars should be sorted from the most to the least profitable

In [4]:
```
import plotly
import plotly.plotly as py
import plotly.graph_objs as go
```

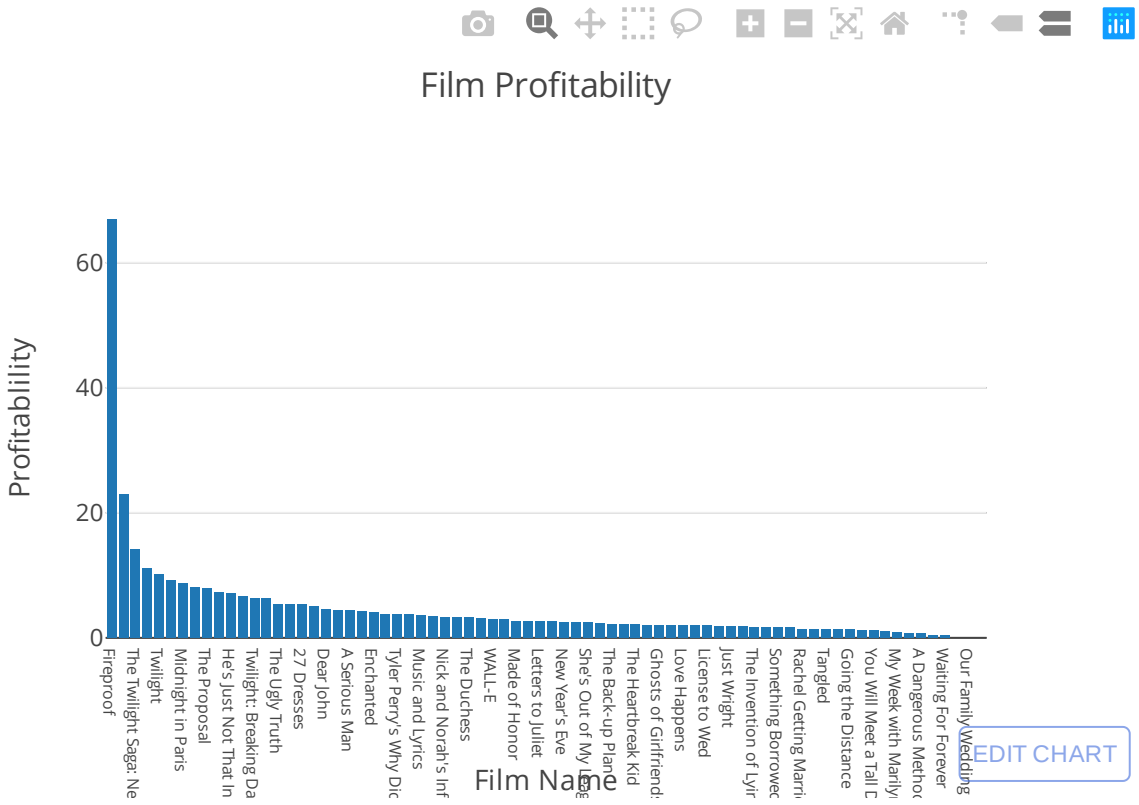
In [5]:
```
hpfp1=hp[["Film","Profitability"]]
hpfp=hpfp1.sort_values(by="Profitability",ascending=False)
```

In [43]:
```
tracea = [go.Bar(
    x=hpfp["Film"],
    y=hpfp["Profitability"]
    )]
layouta = go.Layout(
    title="Film Profitability",
    autosize=False,
    width=600,
    height=400,
    xaxis=dict(title="Film Name",
               tickfont=dict(
              size=8
        )),
    yaxis=dict(title="Profitablility")
```

```
)
figa = go.Figure(data=tracea,layout=layouta)
py.iplot(figa, filename="basic-bar")
```

```
D:\Downloadsss\Anaconda\lib\site-packages\IPython\core\display.py:689: UserW
arning:

Consider using IPython.display.IFrame instead
```

Out[43]:



2.b.A histogram showing the number of films for each Genre. The X axis is the Genre. The Y axis is the number of films in the spreadsheet from each Genre.

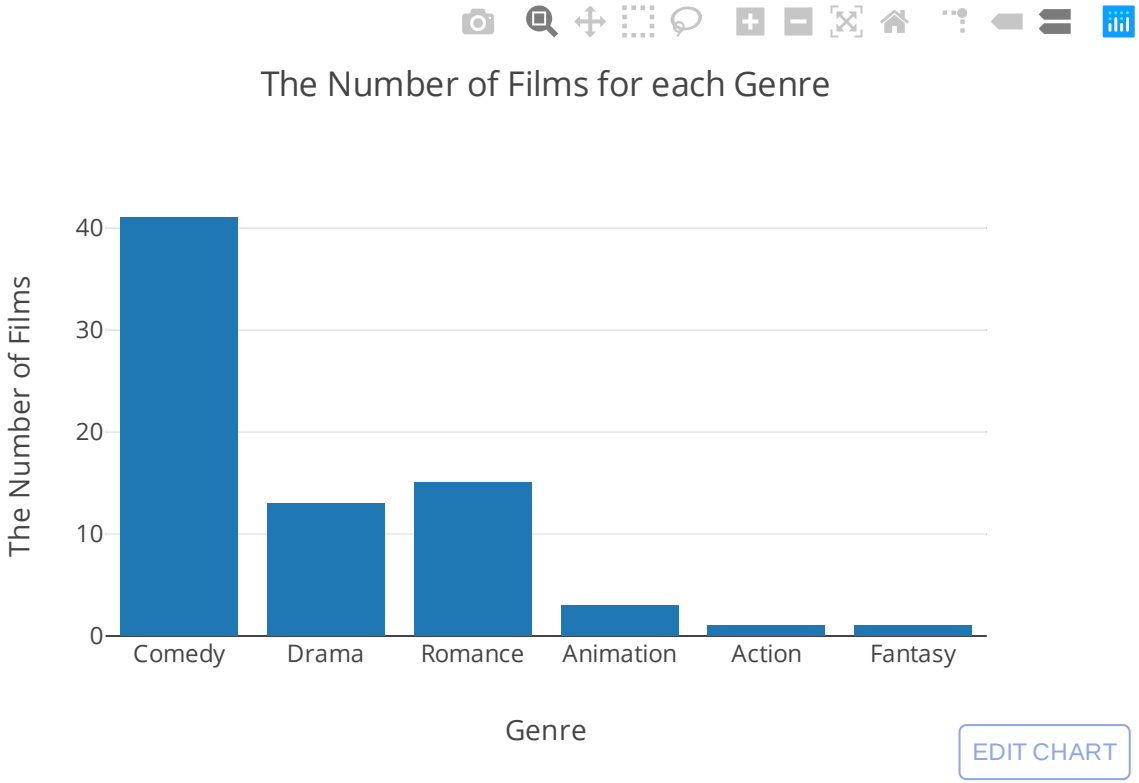
In [44]:

```
traceb = [go.Histogram(
    histfunc = "count",
    x=hp["Genre"],
    y=hp["Film"],
    name = "count"
    )]
layoutb = go.Layout(
```

```
    title="The Number of Films for each Genre",
    autosize=False,
    width=600,
    height=400,
    xaxis=dict(title="Genre"),
    yaxis=dict(title="The Number of Films")
)
figb = go.Figure(data=traceb,layout=layoutb)
py.iplot(figb, filename="basic histogram")
```

```
D:\Downloadsss\Anaconda\lib\site-packages\IPython\core\display.py:689: UserW
arning:

Consider using IPython.display.IFrame instead
```

Out[44]:



2.c.A figure with two bar plots: Worldwide Gross and Audience. The X axis is the film index. When the mouse cursor hovers over a bar, the film's title should be displayed in the tooltip.

In [45]:

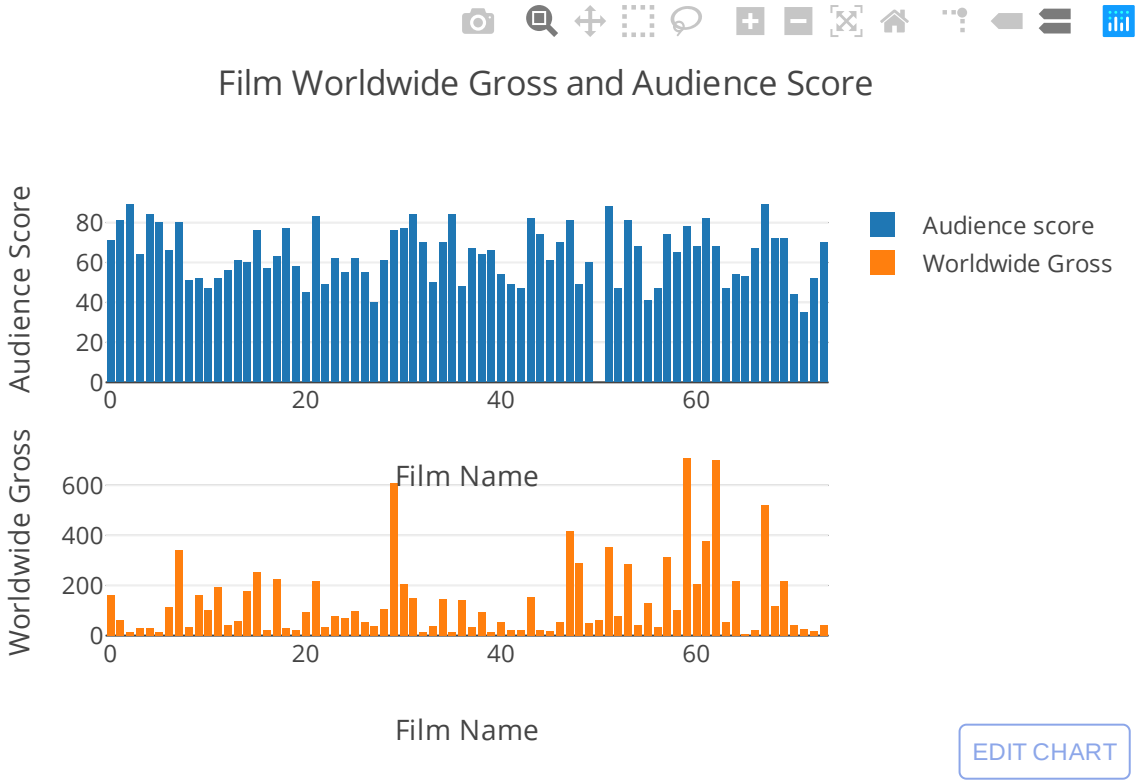
```
from plotly import tools
```

```
x=hp["Film"]

tracec1 = go.Bar(
    x=x.index,
    y=hp["Audience  score %"],
    name="Audience score",
    text=x
)
tracec2 = go.Bar(
    x=x.index,
    y=hp["Worldwide Gross"],
    name="Worldwide Gross",
    text=x
)

figc=tools.make_subplots(rows=2,cols=1)

figc.append_trace(tracec1,1,1)
figc.append_trace(tracec2,2,1)

figc['layout']['xaxis1'].update(title="Film Name")
figc['layout']['yaxis1'].update(title='Audience Score')
figc['layout']['xaxis2'].update(title='Film Name')
figc['layout']['yaxis2'].update(title='Worldwide Gross')
figc['layout'].update(autosize=False,width=600,height=400,title="Film Worldwid
e Gross and Audience Score")

py.iplot(figc, filename="subplot-bar")
```

```
This is the format of your plot grid:
[ (1,1) x1,y1 ]
[ (2,1) x2,y2 ]
```

```
D:\Downloadsss\Anaconda\lib\site-packages\IPython\core\display.py:689: UserW
arning:

Consider using IPython.display.IFrame instead
```

Out[45]:



2.d.A line chart showing the profitability of the films over the years. The X axis is the Year. The Y axis is the profitability.

In [46]:
```
hpd=pd.pivot_table(hp,values="Profitability",index="Year",aggfunc="mean")

traced = go.Scatter(
    x = hpd.index,
    y = hpd.values,
    mode="lines"
)

datad=[traced]

layoutd = go.Layout(
    title="Yearly Profitability",
    autosize=False,
    width=600,
    height=400,
    xaxis=dict(title="Year",
```

```
                type = "category"),
    yaxis=dict(title="Avg of Profitability")
)

figd = go.Figure(data=datad, layout=layoutd)
py.iplot(figd, filename="basic-line")
```

```
High five! You successfully sent some data to your account on plotly. View y
our plot in your browser at https://plot.ly/~rfang1/0 or inside your plot.ly
account where it is named 'basic-line'
```

```
D:\Downloadsss\Anaconda\lib\site-packages\IPython\core\display.py:689: UserW
arning:

Consider using IPython.display.IFrame instead
```

Out[46]:

2.e.A dot plot showing the Rotten Tomato %. The X axis is the Rotten Tomato %. The Y axis is the film title.

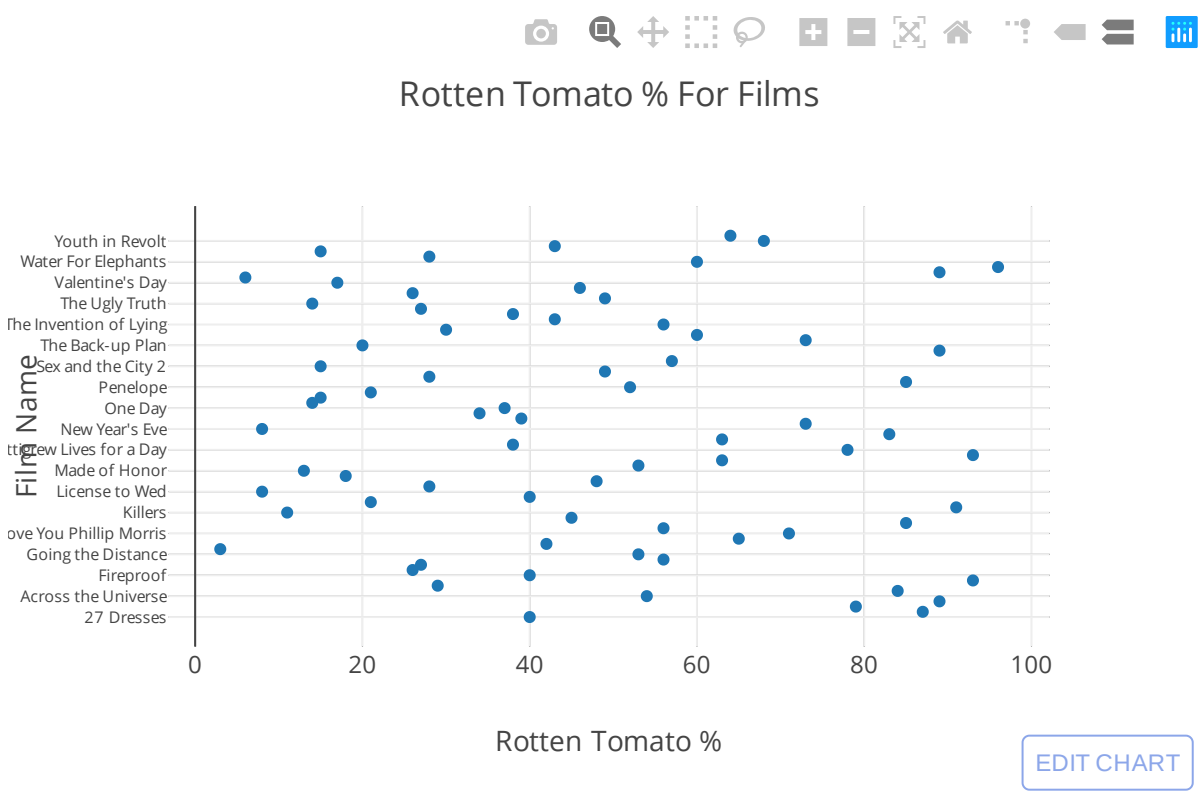
In [47]:

```
tracee = go.Scatter(
```

```
    x = hp["Rotten Tomatoes %"],
    y = hp["Film"],
    mode="markers"
)

datae=[tracee]

layoute = go.Layout(
    title="Rotten Tomato % For Films",
    autosize=False,
    width=600,
    height=400,
    xaxis=dict(title="Rotten Tomato %"),
    yaxis=dict(title="Film Name",
               tickfont=dict(
              size=8)
))

fige = go.Figure(data=datae, layout=layoute)
py.iplot(fige, filename="basic-dot")
```

```
D:\Downloadsss\Anaconda\lib\site-packages\IPython\core\display.py:689: UserW
arning:

Consider using IPython.display.IFrame instead
```

Out[47]:



3.Load Housing_price.csv and use Plotly to create the following charts. Every figure must include a title. Each axis must be labelled.

In [13]:
```
hspr = pd.read_csv("Housing_price.csv")
```

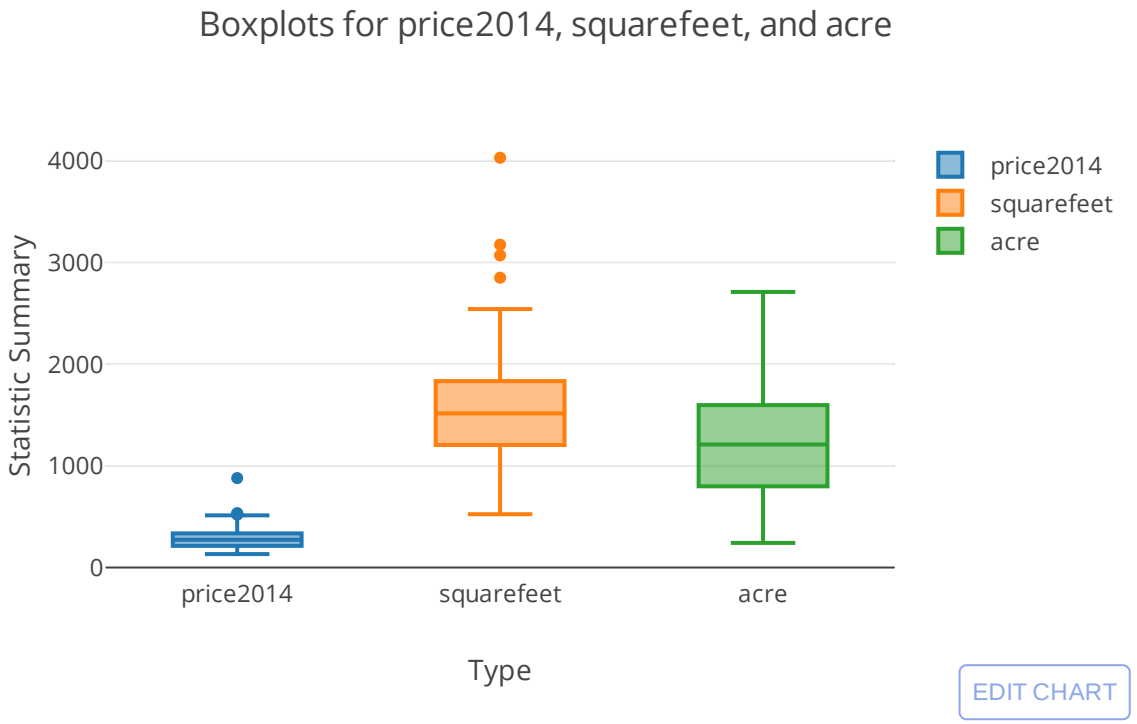
3.a.A figure that contains three boxplots: price2014, squarefeet, and acre.

In [48]:
```
trace3a1 = go.Box(
    y=hspr["price2014"],
    name ="price2014"
)
trace3a2 = go.Box(
    y=hspr["squarefeet"]*1000,
    name ="squarefeet"
)
trace3a3 = go.Box(
    y=hspr["acre"]*4840,
    name ="acre"
)
```

```
data3a = [trace3a1,trace3a2,trace3a3]

layout3a = go.Layout(
    title="Boxplots for price2014, squarefeet, and acre",
    autosize=False,
    width=600,
    height=400,
    xaxis=dict(title="Type"),
    yaxis=dict(title="Statistic Summary")
)
fig3a = go.Figure(data=data3a, layout=layout3a)
py.iplot(fig3a,filename="basic-boxplot")
```

```
D:\Downloadsss\Anaconda\lib\site-packages\IPython\core\display.py:689: UserW
arning:

Consider using IPython.display.IFrame instead
```

Out[48]:

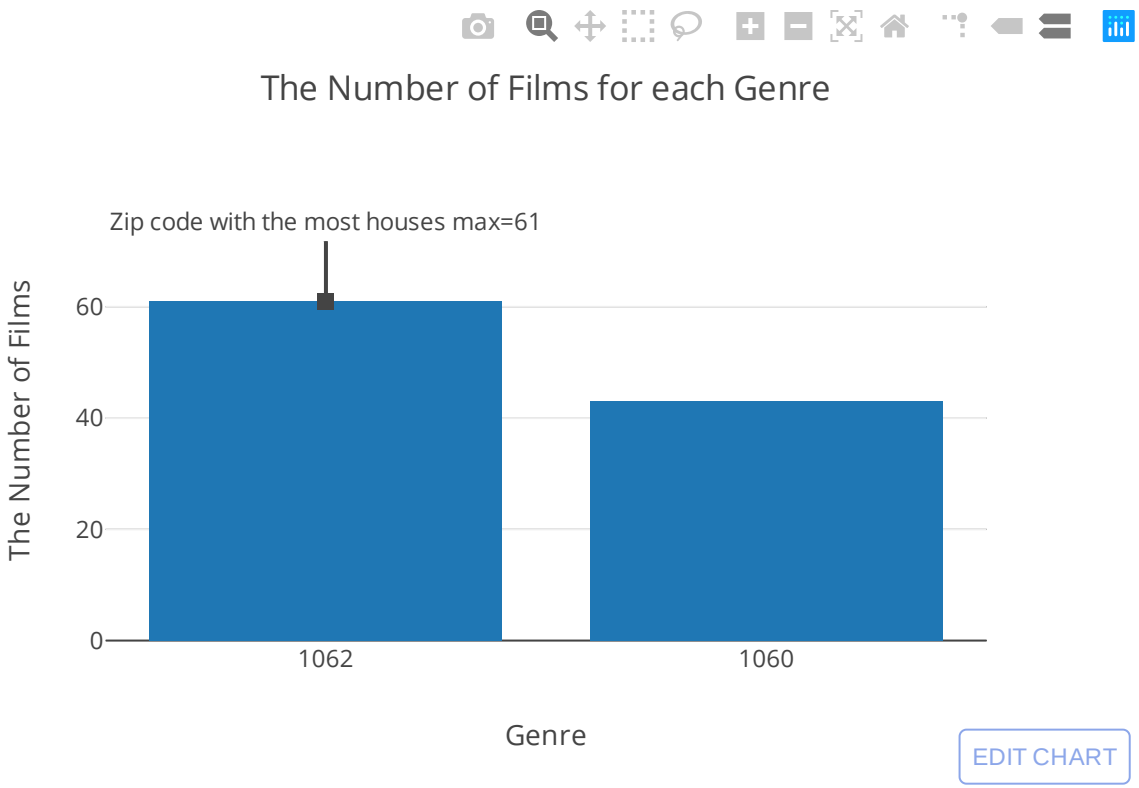


3.b.A histogram showing the number of houses per Zip code. The X axis is the Zip codes. Create an annotation pointing to the Zip code with the most houses.

In [49]:
```
trace3b = [go.Histogram(
    histfunc = "count",
    x=hspr["zip"],
    y=hspr["housenum"],
    name = "count"
    )]
layout3b = go.Layout(
    title="The Number of Films for each Genre",
    autosize=False,
    width=600,
    height=400,
    xaxis=dict(title="Genre",
               type = "category"),
    yaxis=dict(title="The Number of Films"),
    annotations=[
        dict(
            x="1062",
            y="61",
            xref='x',
            yref='y',
            text='Zip code with the most houses max=61',
            showarrow=True,
            arrowhead=7,
            ax=0,
            ay=-40
        )]
)


fig3b = go.Figure(data=trace3b,layout=layout3b)
py.iplot(fig3b, filename="basic histogram2")
```

```
D:\Downloadsss\Anaconda\lib\site-packages\IPython\core\display.py:689: UserW
arning:

Consider using IPython.display.IFrame instead
```

Out[49]:



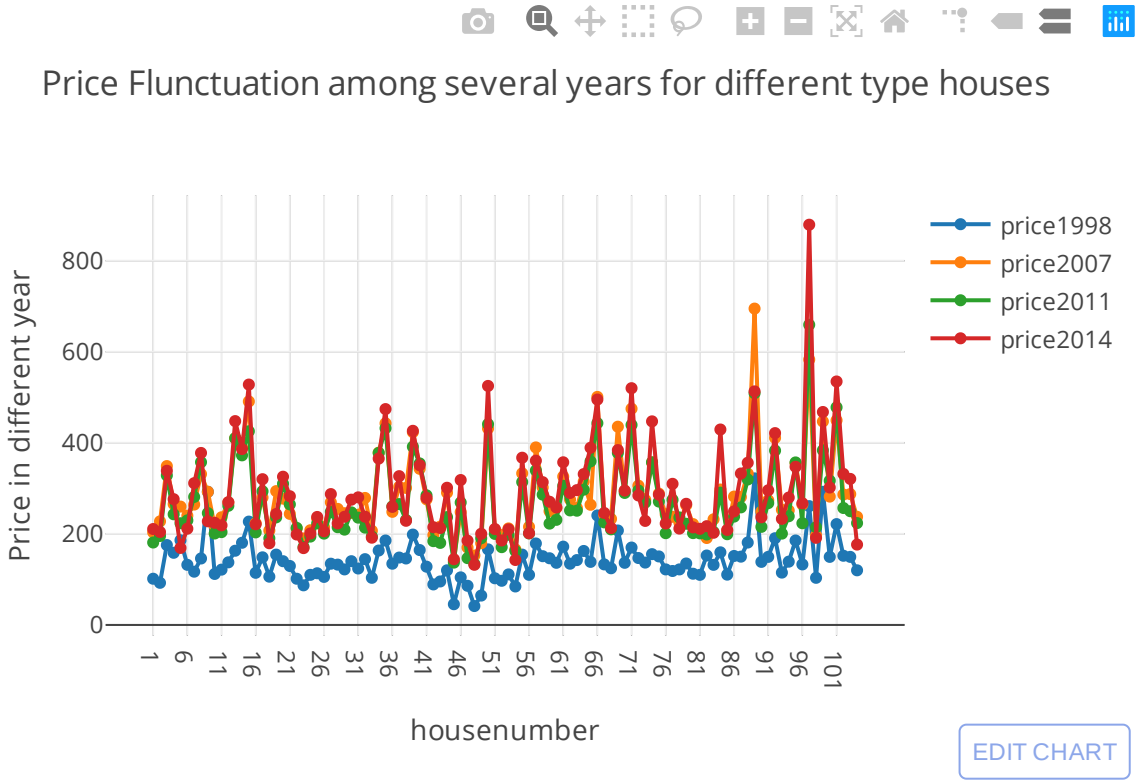
3.c.A line chart with four lines: price1998, price2007, price2011, and price2014. The X axis is the house num. The Y axis is the value from the four columns listed above. When the mouse cursor hovers over a marker, the street address of the house should be displayed in the tooltip.

In [50]:
```
hspr["Address"] = hspr["streetno"].map(str)+" "+hspr["streetname"].map(str)

trace3c1 = go.Scatter(
    x = hspr["housenum"],
    y = hspr["price1998"],
    mode = 'lines+markers',
    name = 'price1998',
    text=hspr['Address']
)
trace3c2 = go.Scatter(
    x = hspr["housenum"],
    y = hspr["price2007"],
    mode = 'lines+markers',
    name = 'price2007',
    text=hspr['Address']
)
```

```
trace3c3 = go.Scatter(
    x = hspr["housenum"],
    y = hspr["price2011"],
    mode = 'lines+markers',
    name = 'price2011',
    text=hspr['Address']
)
trace3c4 = go.Scatter(
    x = hspr["housenum"],
    y = hspr["price2014"],
    mode = 'lines+markers',
    name = 'price2014',
    text=hspr['Address']
)
data3c = [trace3c1, trace3c2, trace3c3,trace3c4]
layout3c = go.Layout(
    title="Price Flunctuation among several years for different type houses",
    autosize=False,
    width=600,
    height=400,
    xaxis=dict(title="housenumber",
               type = "category"),
    yaxis=dict(title="Price in different year")
)

fig3c = go.Figure(data=data3c, layout=layout3c)

py.iplot(fig3c, filename='line-mode')
```

```
D:\Downloadsss\Anaconda\lib\site-packages\IPython\core\display.py:689: UserW
arning:

Consider using IPython.display.IFrame instead
```

Out[50]:



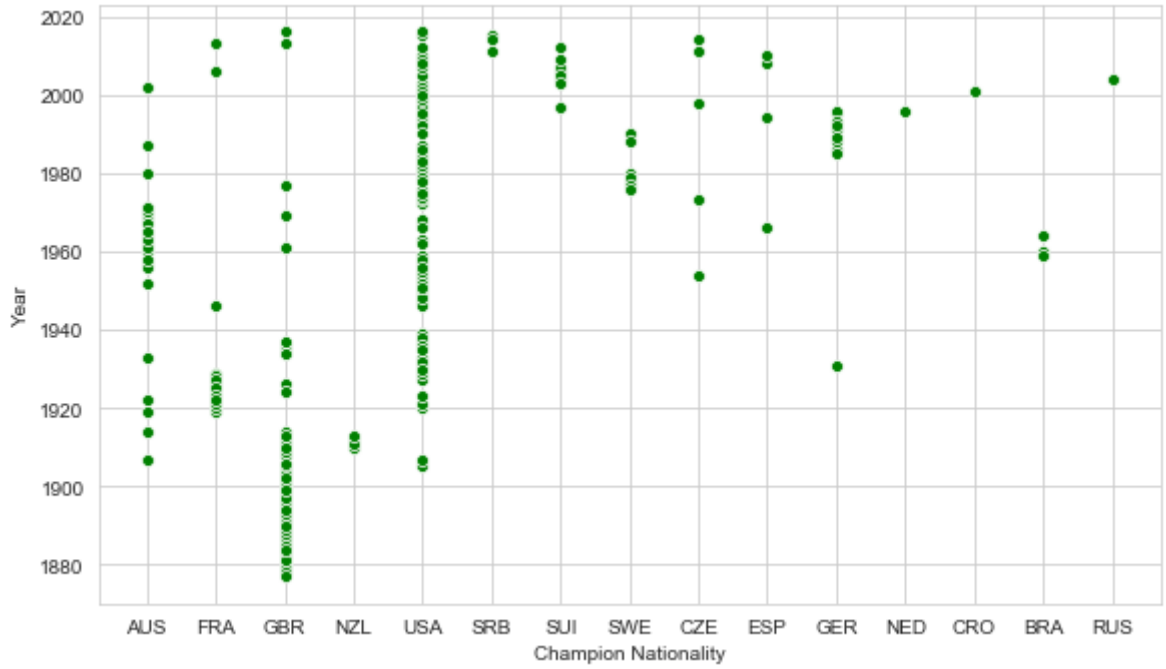
4.Load wimbledons_champions.csv and use Seaborn to create the following charts.

In [25]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style("whitegrid")
wc = pd.read_csv("wimbledons_champions.csv")
```

4.a.A scatterplot showing when a player from different countries won the championship. The X axis is the country. The Y axis is the year. Each circle/dot indicates a player from certain country won the championship in a certain year. The circles should be filled with green color.

In [51]:
```
f, ax = plt.subplots(figsize=(9.5, 5.5))
fig4a=sns.scatterplot(x="Champion Nationality", y="Year",
                      color="green",
                      data=wc)
fig4a
```

Out[51]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x198293b0a58>
```

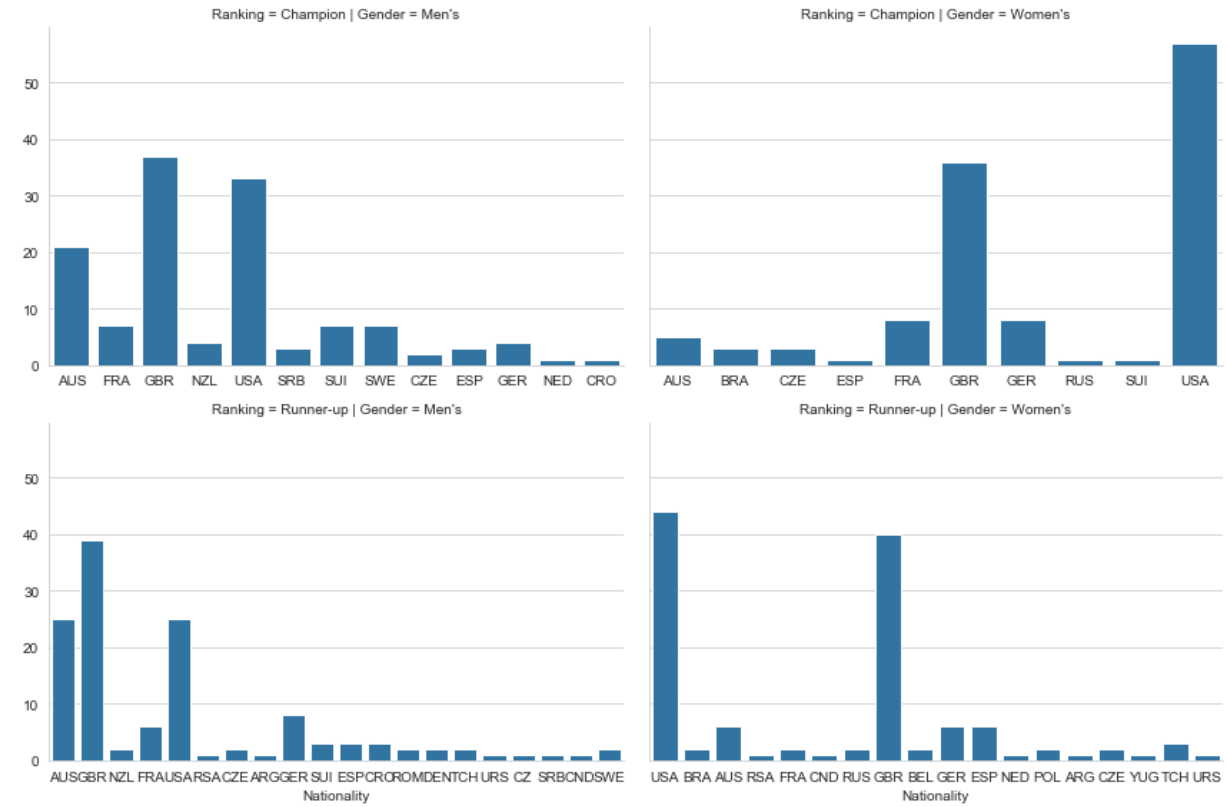


b. Create a grid with four cells. In the first row, show two charts: a histogram showing the number of men's champions for different countries and histogram showing the number of women's champions for different countries. In the second row, show two charts: a histogram showing the number of men's runners-up for different countries and histogram showing the number of women's runners-up for different countries.

In [53]:
```
wc["Runner-up Nationality"] = wc["Runner-up Nationality"].fillna(wc["Runner-up Nationality (Men's)"])
wc["Runner-up"] = wc["Runner-up"].fillna(wc["Runner-Up"])

wc1=pd.DataFrame({"Gender":wc["Gender"],"Nationality":wc["Champion Nationality"],"Ranking":"Champion"})
wc2=pd.DataFrame({"Gender":wc["Gender"],"Nationality":wc["Runner-up Nationality"],"Ranking":"Runner-up"})
wc4b=pd.concat([wc1,wc2])
g = sns.FacetGrid(wc4b, col="Gender", row="Ranking",height=4,aspect=1.5,sharex=False)
g.map(sns.countplot,"Nationality")

plt.show()
plt.savefig("wc4b.png")
```

```
D:\Downloadsss\Anaconda\lib\site-packages\seaborn\axisgrid.py:715: UserWarni
```

```
ng:

Using the countplot function without specifying `order` is likely to produce
an incorrect plot.
```




```
<Figure size 432x288 with 0 Axes>
```

In [ ]: