Assignment_2_solutions

October 11, 2021

[11]:
```
import pandas as pd

# Data Visualization
import seaborn as sns
import matplotlib.pyplot as plt
```

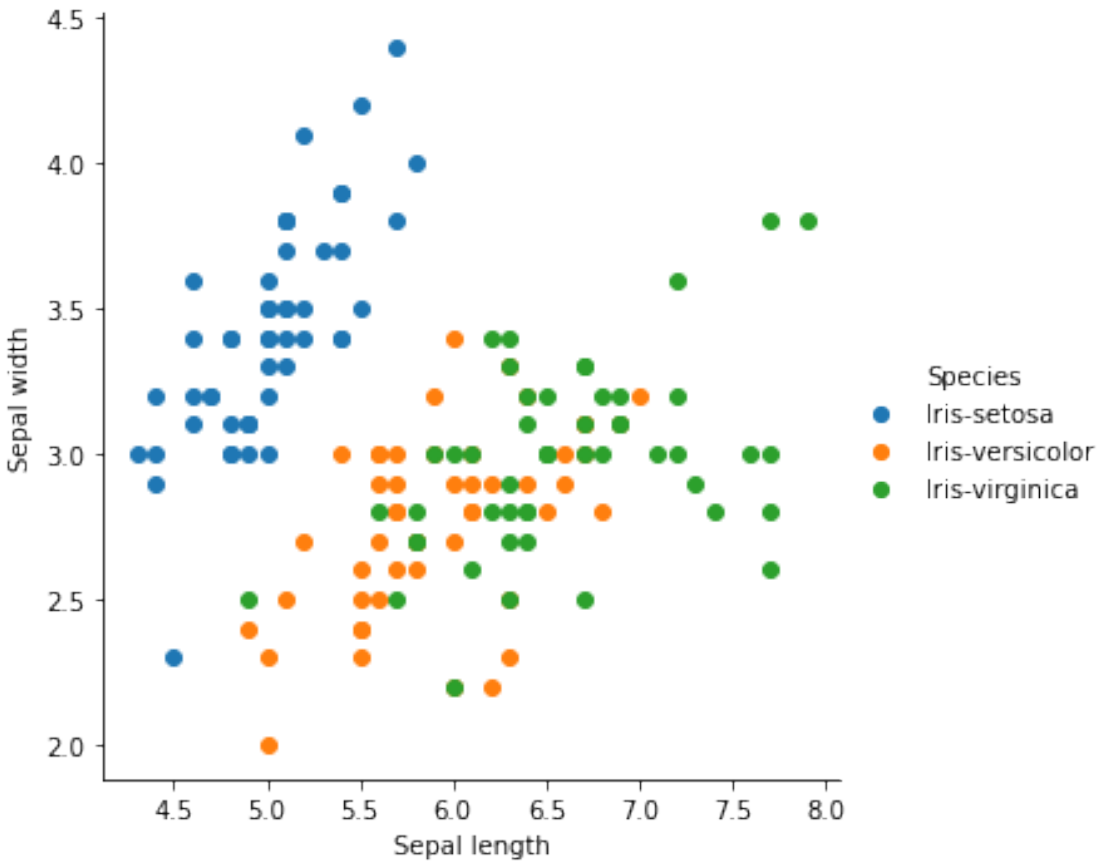
1 I

1.1 I-1

[12]:
```
iris = pd.read_csv('iris.data')
```

[14]:
```
a = sns.FacetGrid(iris, hue = 'Species', height=5)
a.map(plt.scatter, 'Sepal length','Sepal width')
a.add_legend()
```

[14]:
```
<seaborn.axisgrid.FacetGrid at 0x20c68ed78b0>
```

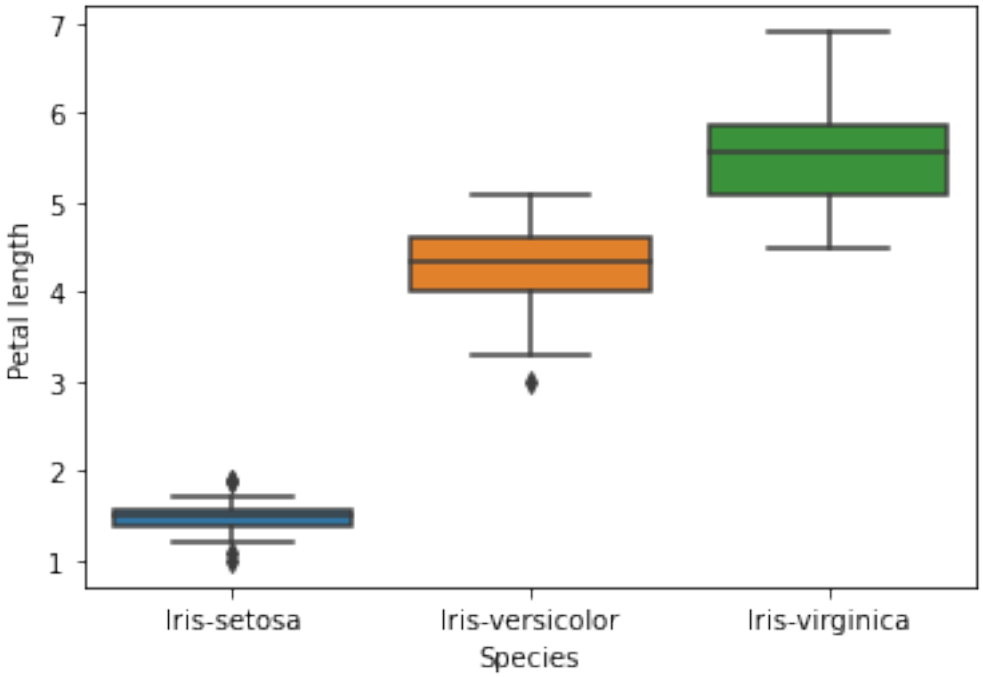


The sepal-length width combination of **Setosa** is quite distinct from that of the other two species.

1.2 I-2

[6]:
```
sns.boxplot(x='Species', y='Petal length', data=iris)
```

[6]:
```
<AxesSubplot:xlabel='Species', ylabel='Petal length'>
```



a) Verginica does not have any outliying values of petal length

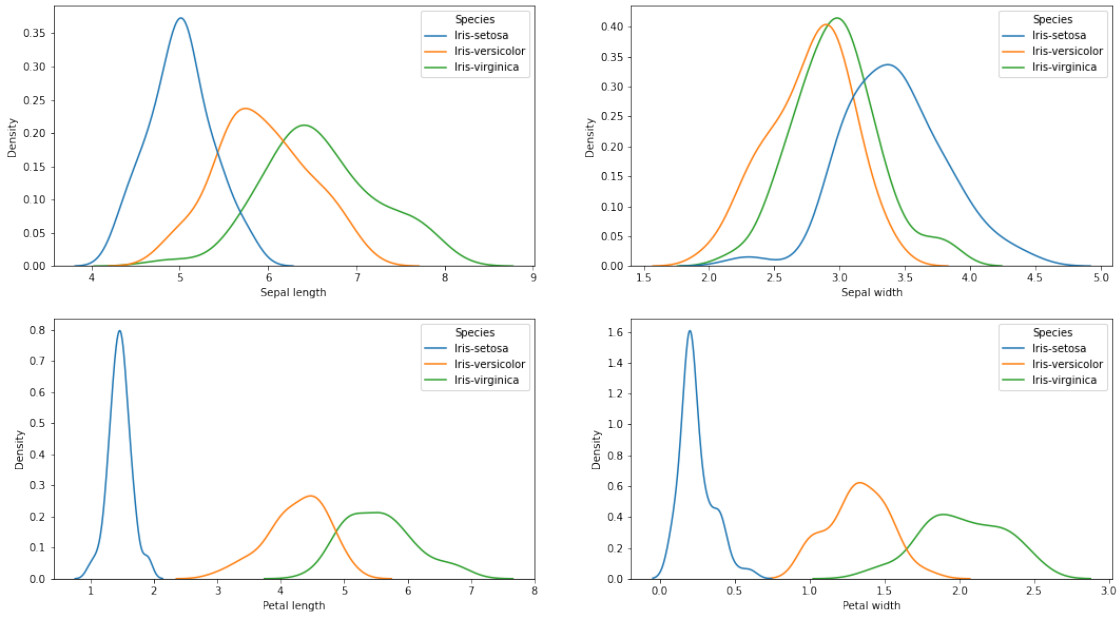
b) Both versiocolor and virginica always have higher petal length than setosa

c) Setosa has the lowest variance of petal length

1.3 II-3

[15]:
```
fig, axes = plt.subplots(2, 2, figsize=(18, 10))
sns.kdeplot(data = iris, x="Sepal length", hue="Species", ax = axes[0,0])
sns.kdeplot(data = iris, x="Sepal width", hue="Species", ax = axes[0,1])
sns.kdeplot(data = iris, x="Petal length", hue="Species", ax = axes[1,0])
sns.kdeplot(data = iris, x="Petal width", hue="Species", ax = axes[1,1])
```

[15]:
```
<AxesSubplot:xlabel='Petal width', ylabel='Density'>
```



a) Densities of versicolor and virginica have the highest overlap in case of Sepal width

[16]:
```
from scipy import stats
data1 = iris[iris['Species']=='Iris-virginica']
data2 = iris[iris['Species']=='Iris-versicolor']
stats.ttest_ind(data1['Sepal width'], data2['Sepal width'])
```

[16]:
```
Ttest_indResult(statistic=3.2057607502218186, pvalue=0.0018191004238894803)
```

b) Yes, they do differ signficantly as p-value $< 5\%$

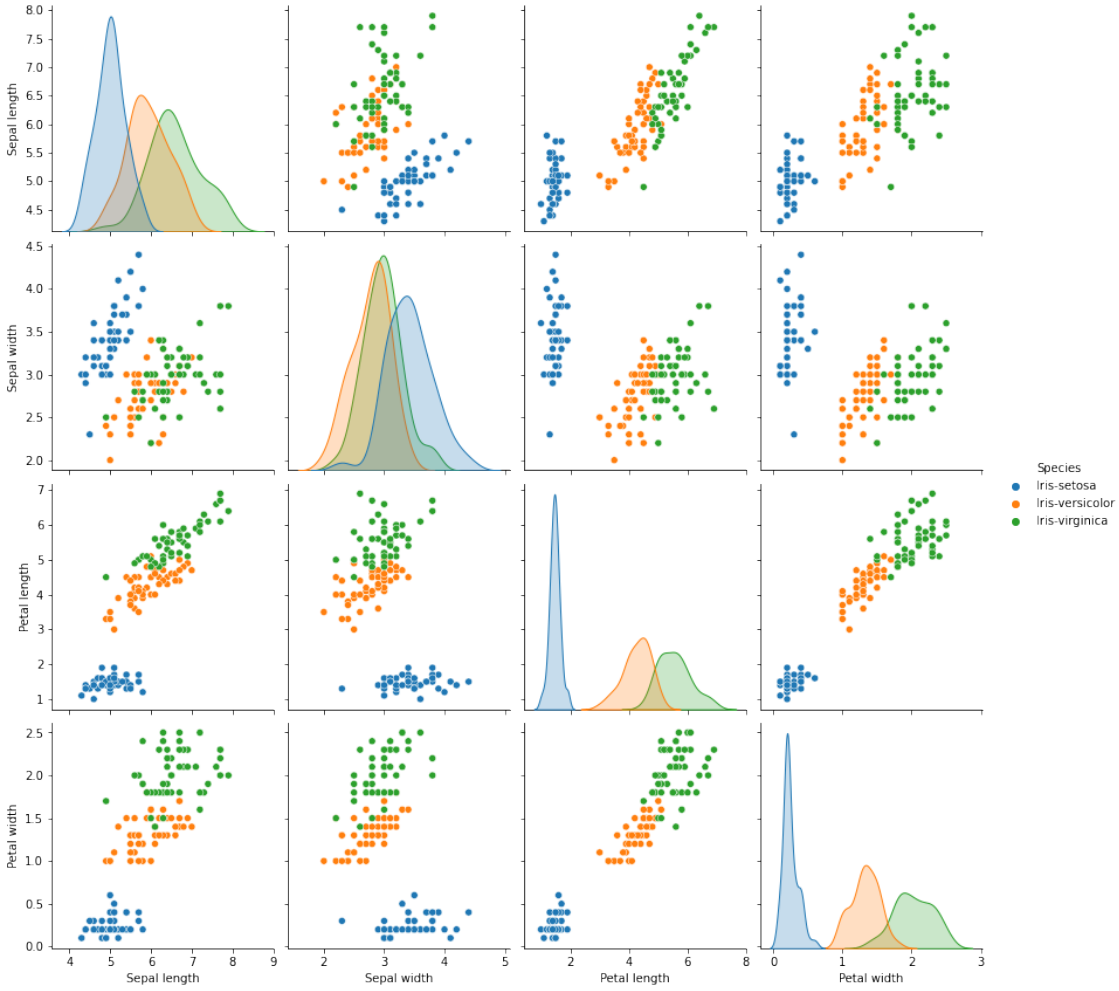
c) Setosa wil be the easiest to identify. Its petal width and petal length seems to be always smaller than those of the other two species. Thus, it can be easily identified by either of these two descriptors.

1.4 II-4

[17]:
```
sns.pairplot(iris.iloc[:,1:], hue='Species', size=3)
```

```
C:\Users\akl0407\Anaconda3\lib\site-packages\seaborn\axisgrid.py:1969:
UserWarning: The `size` parameter has been renamed to `height`; please update
your code.
  warnings.warn(msg, UserWarning)
```

[17]:
```
<seaborn.axisgrid.PairGrid at 0x20c68e70ee0>
```



a) Petal width, as their distribution overlaps the least

b) Sepal width, as their distribution overlaps the most

c) Setosa has the least correlation among its sepal and petal descriptors. Consider the plot between sepal length and petal width. As sepal length is increasing, petal length seems to be more or less constant for setosa. The other 7 subplots with sepal and petal parameters show a similar trend for Setosa.

2 II

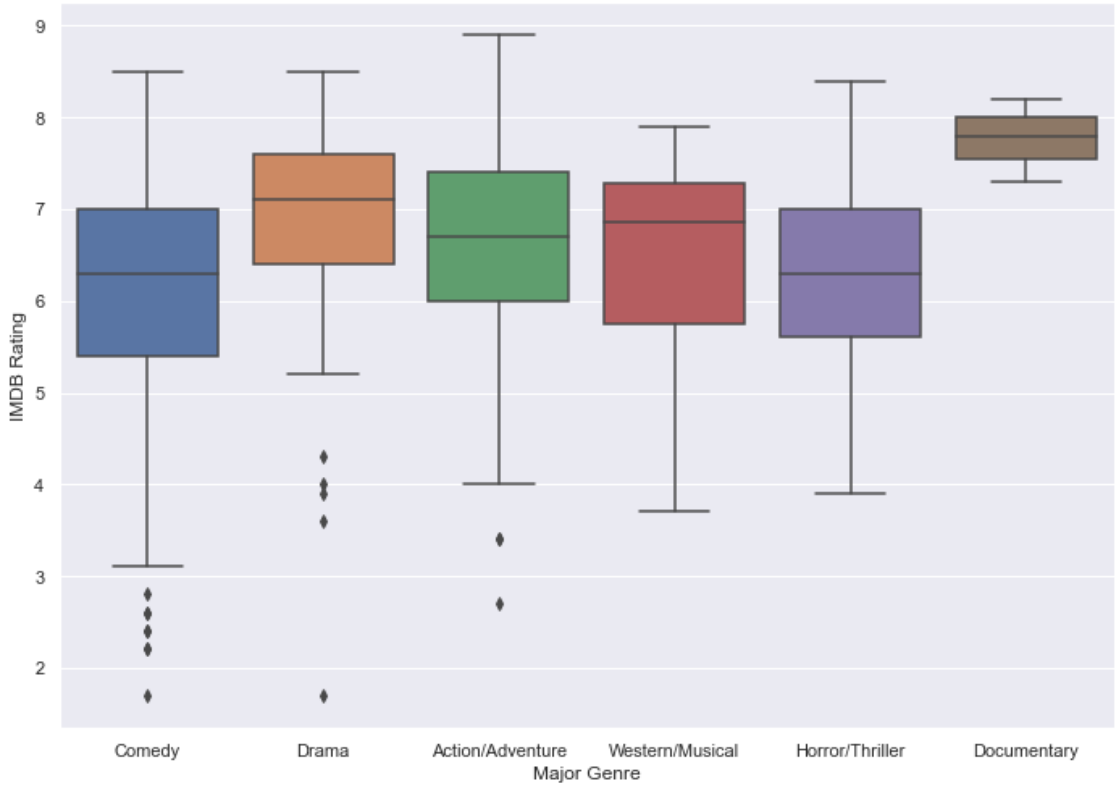
[18]:
```
data = pd.read_csv('Movies.csv')
```

[19]:
```
data=data.dropna()
```

3 II-1

[18]:
```
sns.set(rc={'figure.figsize':(11.7,8.27)})
sns.boxplot(x="Major Genre", y="IMDB Rating", data=data,width=0.8)
```

[18]:
```
<AxesSubplot:xlabel='Major Genre', ylabel='IMDB Rating'>
```



a) I would choose to make a documentary because its IMDB rating is always higher than 7, while with all other genres the rating may go below 7.

b) I would choose to make an action/adventure movie because it has the highest maximum. Thus, there is a possibility of making a movie with the highest IMDB rating.

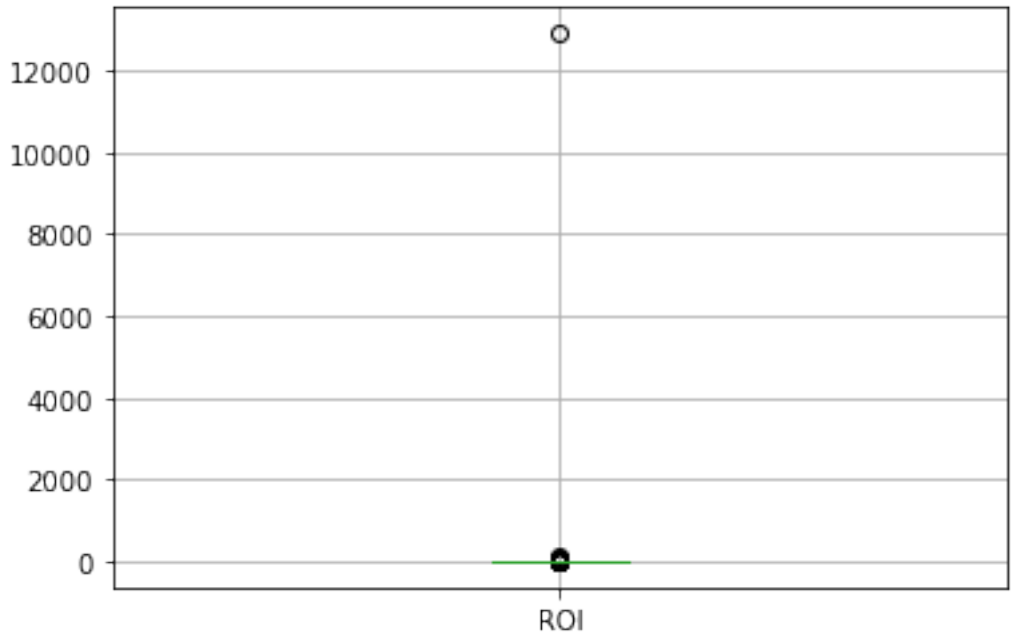
c) Drama, Action/Adventure, and documentary. *Note: Since Action/Adventure is at borderline, the answer is correct without it too!*

3.1 II-2

[21]:
```
data['ROI'] = (data['Worldwide Gross']-data['Production Budget'])/
↪data['Production Budget']
```

[26]:
```
data.boxplot('ROI')
```

[26]: <AxesSubplot:>



a) Yes, there are outliers with respect to ROI

[38]:
```
#Movie corresponding to the most extreme outlier
data.iloc[data['ROI'].argmax(),:2]
```

[38]:
```
Unnamed: 0                   2526
Title         Paranormal Activity
Name: 2474, dtype: object
```

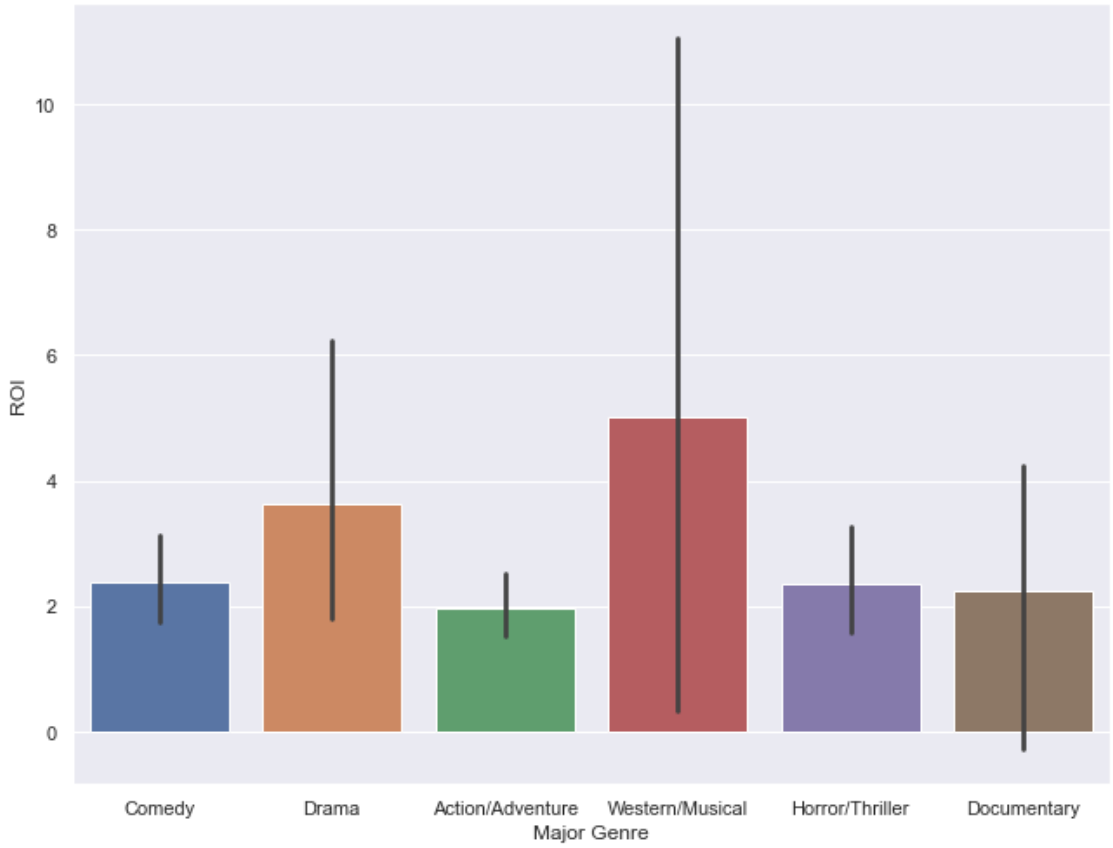
The most extreme outlier corresponds to Paranormal activity.

[41]:
```
#Removing the most extreme outlier from the data
data=data[data['Title']!='Paranormal Activity']
```

b)

[50]:
```
sns.set(rc={'figure.figsize':(10.7,8.27)})
sns.barplot(data = data,x = 'Major Genre',y = 'ROI')
```

[50]:
```
<AxesSubplot:xlabel='Major Genre', ylabel='ROI'>
```



c) I will choose Western/Musical because its 95% confidence interval includes the highest ROI. Thus, there is a possibility of achieving the highest possible ROI.

d) I will choose Drama because the minimum of its 95% ROI confidence interval is higher than the minimum of the 95% ROI confidence interval for other genres. Thus, in the worst case, the ROI will be higher than corresponding worst cases of other genres.

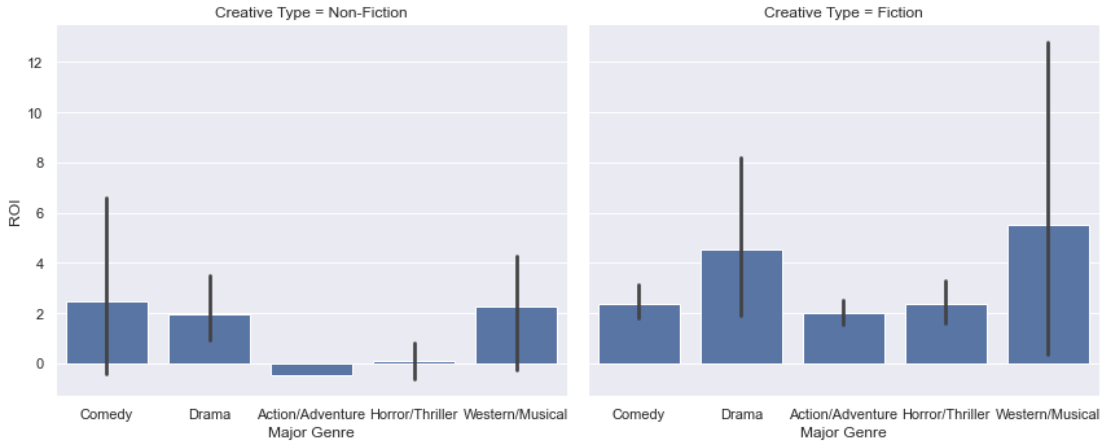
e) I will discard documentary because its 95% confidence interval includes negative ROI. Thus, there are documentries that have incurred a loss.

f)

[51]:
```
a = sns.FacetGrid(data=data,height = 5,aspect = 1.2,col = "Creative␣
↪Type",col_wrap=2)
a.map(sns.barplot,"Major Genre","ROI")
a.add_legend()
```

```
C:\Users\akl0407\Anaconda3\lib\site-packages\seaborn\axisgrid.py:643:
UserWarning: Using the barplot function without specifying `order` is likely to
produce an incorrect plot.
  warnings.warn(warning)
```

[51]: <seaborn.axisgrid.FacetGrid at 0x20c6c5045b0>



g) Among the non-fiction movies, comedy, action/adventure, horror/thriller and western/musical movies may have a negative ROI, as their confidence interval includes negative ROI values. Fiction movies always have a positive ROI.

3.2 II-3

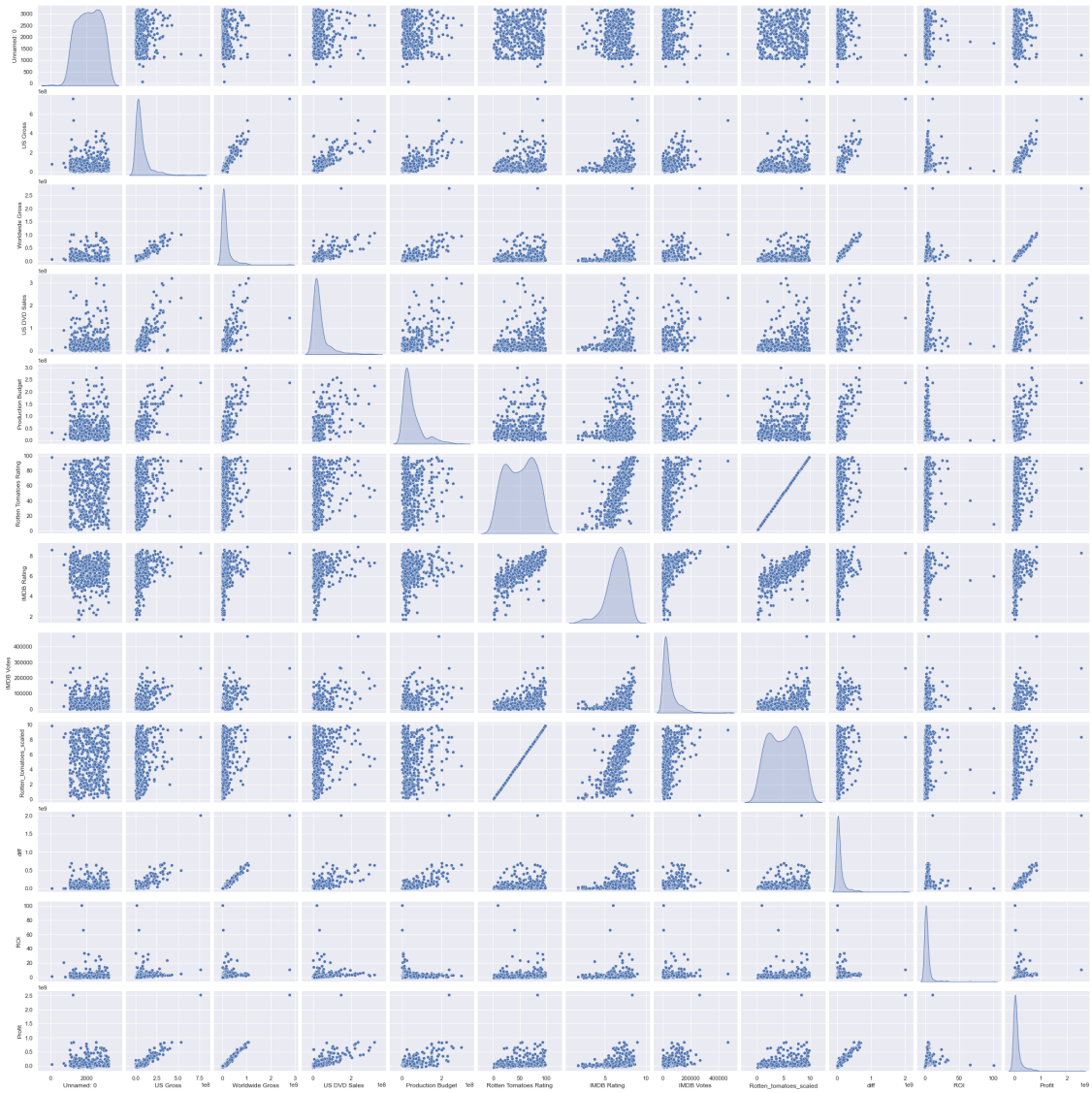
a)

[52]:
```
data['Profit'] = data['Worldwide Gross']-data['Production Budget']
```

b)

[53]:
```
sns.pairplot(data,diag_kind = 'kde')
```

[53]: <seaborn.axisgrid.PairGrid at 0x20c6c98c970>

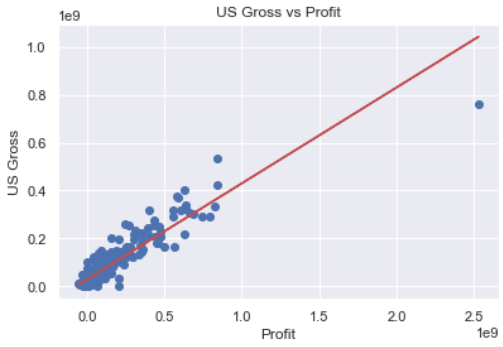


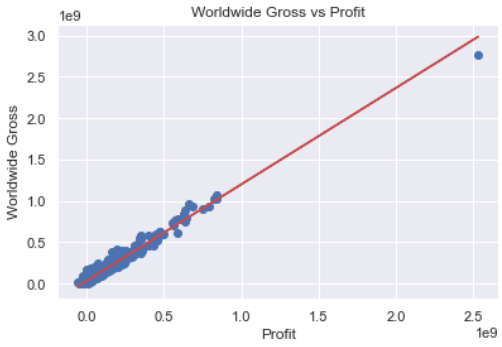
Four variables positively correlated with profit are: US Gross, Worldwide Gross, Production Budget, and US DVD sales

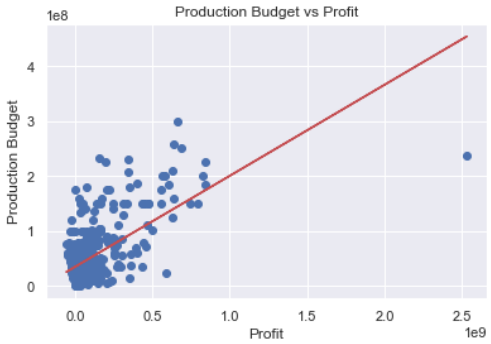
[60]:
```
fig, axes = plt.subplots(2,2,figsize=(15,10))
plt.subplots_adjust(wspace=0.4,hspace=0.5)
x=data['Profit']
y=data['US Gross']
axes[0,0].plot(x,y,'o',color='b')
axes[0,0].set_xlabel('Profit')
axes[0,0].set_ylabel('US Gross')
axes[0,0].set_title('US Gross vs Profit')
z = np.polyfit(x, y, 1)
p = np.poly1d(z)
```

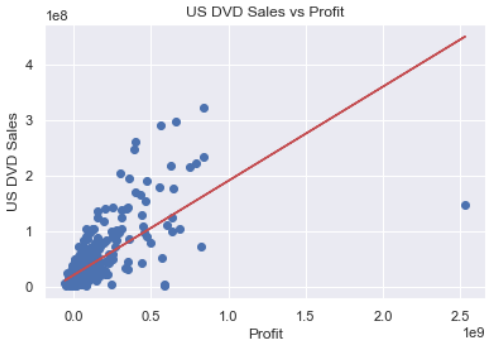
```
#Plot a trendline
axes[0,0].plot(x,p(x),color='r')

y=data['Worldwide Gross']
axes[0,1].plot(x,y,'o',color='b')
axes[0,1].set_xlabel('Profit')
axes[0,1].set_ylabel('Worldwide Gross')
axes[0,1].set_title('Worldwide Gross vs Profit')
z = np.polyfit(x, y, 1)
p = np.poly1d(z)

#Plot a trendline
axes[0,1].plot(x,p(x),color='r')

y=data['Production Budget']
axes[1,0].plot(x,y,'o',color='b')
axes[1,0].set_xlabel('Profit')
axes[1,0].set_ylabel('Production Budget')
axes[1,0].set_title('Production Budget vs Profit')
z = np.polyfit(x, y, 1)
p = np.poly1d(z)

#Plot a trendline
axes[1,0].plot(x,p(x),color='r')

y=data['US DVD Sales']
axes[1,1].plot(x,y,'o',color='b')
axes[1,1].set_xlabel('Profit')
axes[1,1].set_ylabel('US DVD Sales')
axes[1,1].set_title('US DVD Sales vs Profit')
z = np.polyfit(x, y, 1)
p = np.poly1d(z)

#Plot a trendline
axes[1,1].plot(x,p(x),color='r')
```

[60]: [<matplotlib.lines.Line2D at 0x20c7428aa60>]









3.3 II-4

a)

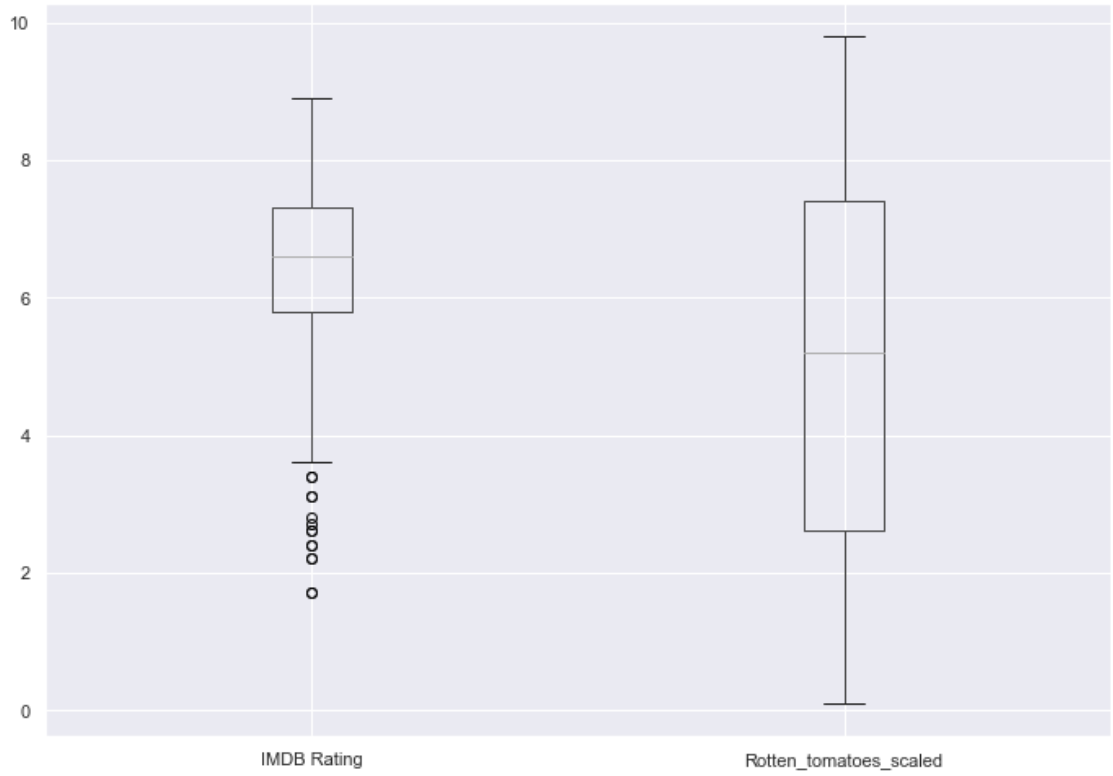
[61]:
```
data['Rotten_tomatoes_scaled'] = data['Rotten Tomatoes Rating']/10
```

Note: As the column *'Rotten_tomatoes_scaled'* already exists in the data, you may just identify this column in the data to get points for this part. You don't need to write code.

b)

[26]:
```
data.boxplot(['IMDB Rating','Rotten_tomatoes_scaled'])
```

[26]: <AxesSubplot:>

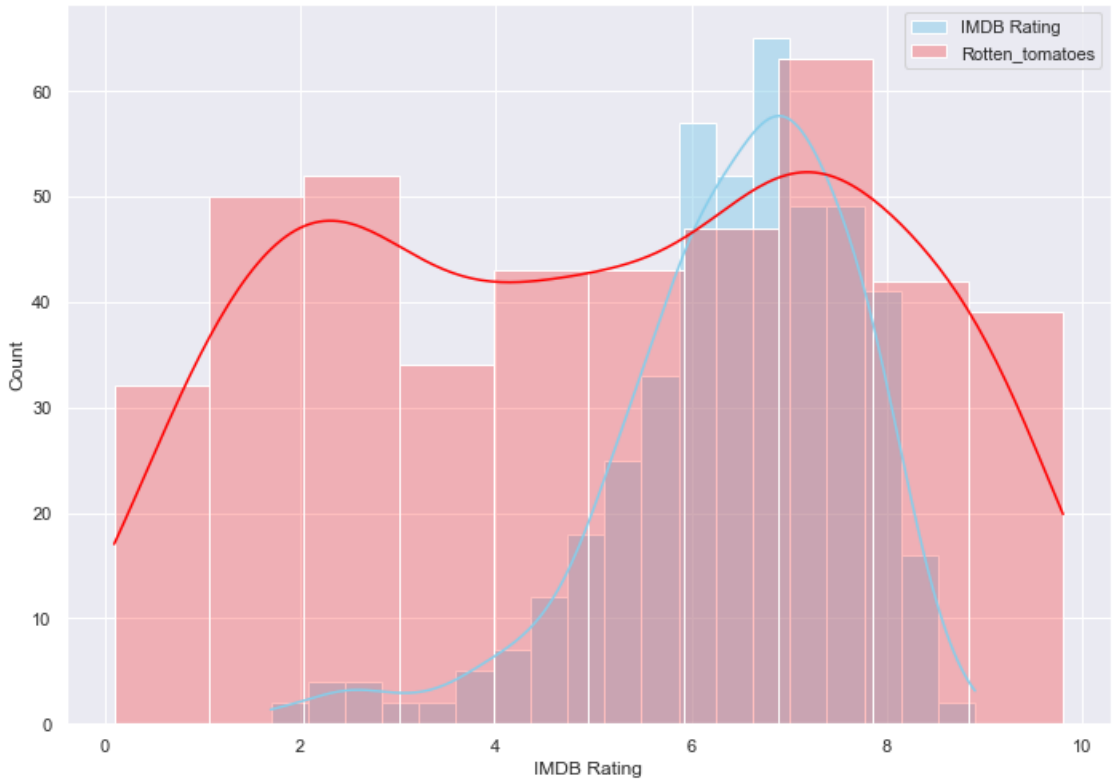


The rotten tomatoes rating distinguishes more among movies, as its variance is larger.

c)

[28]:
```
sns.histplot(data=data, x="IMDB Rating", color="skyblue", label="IMDB Rating",␣
↪kde=True)
sns.histplot(data=data, x="Rotten_tomatoes_scaled", color="red",␣
↪label="Rotten_tomatoes", kde=True,alpha=0.25)
plt.legend()
```

[28]:
```
<matplotlib.legend.Legend at 0x29e2cf93c10>
```



The distribution of Rotten tomatoes rating is bi-modal. The modes are approximately at 2 and 7. Note that any values of the modes in [1,3], and [6,8] is acceptable.

3.4 II-5

a) The plot is already drawn in II-4(c)

b) Yes, as the plot has a negative skew, the mean is skewed to the left.

Question_Gourds2

October 15, 2021

1 Gourds Questions

17 points total

For this part of the assignment, please load and use the data set "GourdData.csv" from Files on Canvas. This data was collected from the US Department of Agriculture's Quick Stats tool (https://quickstats.nass.usda.gov/). Each row includes results from a survey on a given state's gourd production in a given year.

Rows with "NaN" in certain columns indicate that we don't have complete information on a state's crop in a certain year, but be sure to keep all rows when you're working with this data set.

Be sure to format all your visualizations with appropriate titles and axis labels.

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

sns.set()
```

[3]:
```
gourd_df = pd.read_csv("GourdData.csv", index_col=0)
gourd_df['Year'] = gourd_df['Year'].astype(str)
gourd_df.head(3)
```

[3]:
```
   Year       State  PUMPKINS - ACRES HARVESTED  PUMPKINS - ACRES PLANTED  \
0  2000  CALIFORNIA                      5900.0                    5900.0
1  2000     FLORIDA                         NaN                       NaN
2  2000     GEORGIA                         NaN                       NaN

   PUMPKINS - PRICE RECEIVED  SQUASH - ACRES HARVESTED  \
0                       10.6                    8300.0
1                        NaN                   11200.0
2                        NaN                   10500.0

   SQUASH - ACRES PLANTED  SQUASH - PRICE RECEIVED
0                  8300.0                     20.0
1                 11500.0                     32.2
2                 12000.0                     24.8
```

1.0.1 Question 1

(4 points for code incl. visualization, 1 point for answer)

Find the top five pumpkin-producing states in the year 2020. Say we're interested in those five states' pumpkin production from 2016-2020.

Assuming that the amount of pumpkin per acre harvested is constant, produce an appropriate seaborn visualization to meet this goal. Briefly provide at least two observations about your visualization and the US pumpkin market.

[4]:
```
sorted_2020_df = gourd_df[gourd_df["Year"]=="2020"].sort_values(by = ["PUMPKINS␣
↪- ACRES HARVESTED","State"],\
                                                                ascending = False).
↪head(5)
sorted_2020_df
```

[4]:

	Year	State	PUMPKINS - ACRES HARVESTED	PUMPKINS - ACRES PLANTED	\
294	2020	ILLINOIS	15900.0	16400.0	
303	2020	PENNSYLVANIA	7000.0	7300.0	
295	2020	INDIANA	6000.0	6200.0	
298	2020	NEW YORK	5600.0	6300.0	
305	2020	VIRGINIA	5400.0	5500.0	

	PUMPKINS - PRICE RECEIVED	SQUASH - ACRES HARVESTED	\
294	NaN	NaN	
303	NaN	NaN	
295	NaN	NaN	
298	NaN	4200.0	
305	NaN	NaN	

	SQUASH - ACRES PLANTED	SQUASH - PRICE RECEIVED
294	NaN	NaN
303	NaN	NaN
295	NaN	NaN
298	4400.0	NaN
305	NaN	NaN

[5]:
```
top_5_df = gourd_df[gourd_df["Year"] >= "2016"]
top_5_df = top_5_df[top_5_df["State"].
↪isin(["ILLINOIS","PENNSYLVANIA","INDIANA","NEW YORK","VIRGINIA"])]
top_5_df.head(3)
```

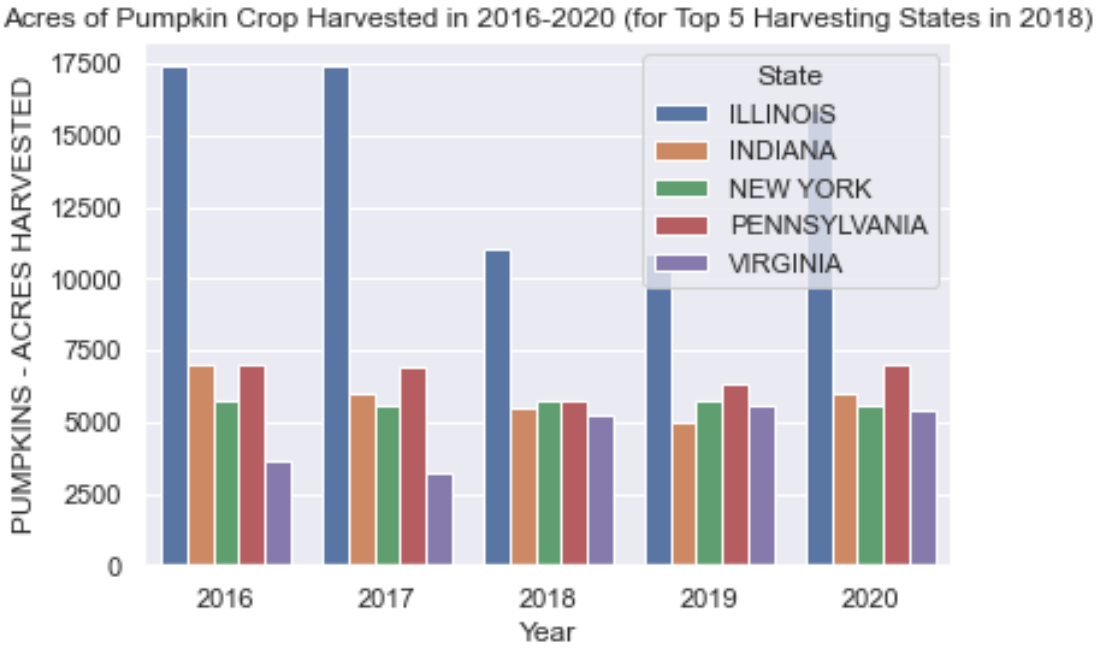
[5]:

	Year	State	PUMPKINS - ACRES HARVESTED	PUMPKINS - ACRES PLANTED	\
227	2016	ILLINOIS	17400.0	18100.0	
228	2016	INDIANA	7000.0	7100.0	
232	2016	NEW YORK	5700.0	6500.0	

	PUMPKINS - PRICE RECEIVED	SQUASH - ACRES HARVESTED	\

```
227                          NaN                          NaN
228                          NaN                          NaN
232                          NaN                       5600.0

     SQUASH - ACRES PLANTED  SQUASH - PRICE RECEIVED
227                     NaN                      NaN
228                     NaN                      NaN
232                  5800.0                      NaN
```

[6]:
```
sns.barplot(x='Year', y='PUMPKINS - ACRES HARVESTED', hue='State',␣
 ↪data=top_5_df)
plt.title("Acres of Pumpkin Crop Harvested in 2016-2020 (for Top 5 Harvesting␣
 ↪States in 2018)")
plt.show()
```



We're just looking for something that proves students can interpret their visualization. Two or more solid observations will work here. For example, full credit for:

I've learned that from 2016-2020 Illinois farmers have harvested the greatest number of pumpkins out of all US states. In 2016, farmers in Illinois harvested more than double the acres of pumpkin harvested by any other state.

1.0.2 Question 2

(3 points for code incl. visualization, 1 point for answer)

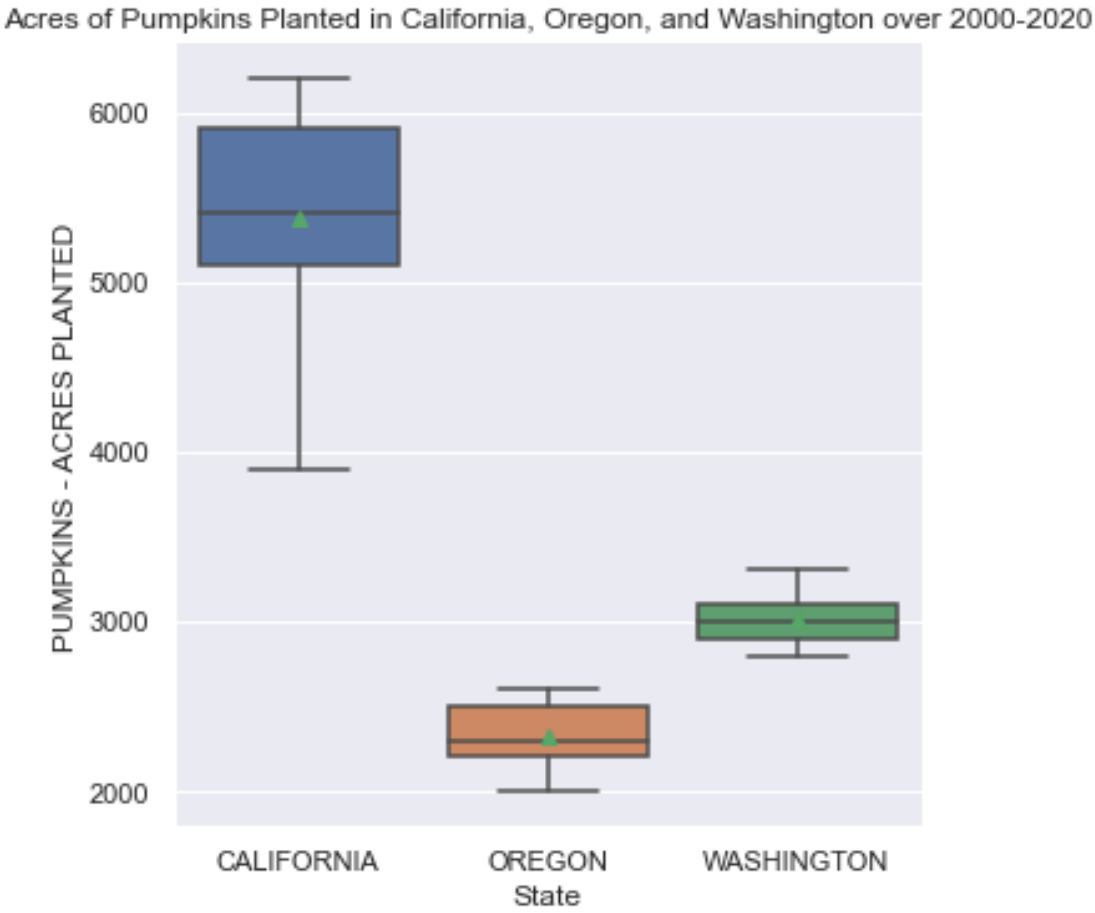
Suppose we're considering Pacific states California, Oregon, and Washington. For each state, we

want to know the mean number of acres of pumpkins planted across 2000-2020. We're also interested in looking at each state's median number of acres of pumpkins planted in the same time horizon.

Produce one appropriate visualization to meet both of these goals. Briefly provide at least one observation about this visualization and the US pumpkin market.

[7]:
```
gourd_pac_df = gourd_df[gourd_df['State'].
 ↪isin(["CALIFORNIA","OREGON","WASHINGTON"])]
sns.factorplot(x='State', y='PUMPKINS - ACRES PLANTED', kind='box',␣
 ↪data=gourd_pac_df, showmeans=True)
plt.title("Acres of Pumpkins Planted in California, Oregon, and Washington over␣
 ↪2000-2020")
plt.show()
```

```
/usr/local/lib/python3.7/site-packages/seaborn/categorical.py:3704: UserWarning:
The `factorplot` function has been renamed to `catplot`. The original name will
be removed in a future release. Please update your code. Note that the default
`kind` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.
  warnings.warn(msg)
```



We're just looking for something that proves students can interpret their visualization. One or more solid observation(s) will work here. For example, full credit for:

The minimum number of acres of pumpkins that California planted over 2000-2020 is greater than the maximum number of acres of pumpkins that Oregon and Washington planted in the same time period.

1.0.3 Question 3

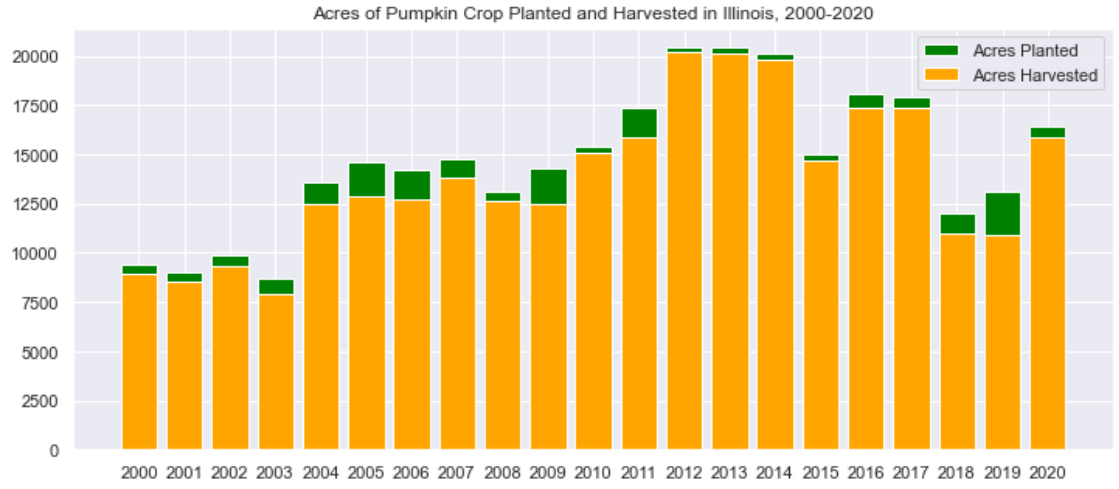
(3 points for code incl. visualization, 1 point for answer)

Consider the acres of pumpkins planted and acres of pumpkins harvested in Illinois from 2000-2020. Are there any trends in the number of acres planted and harvested over time? Produce a visualization to answer this question, along with a short written answer.

Hint: If you're having trouble with label sizes, play around with running: *sns.set(rc = {'figure.figsize' : (30, 30), 'axes.labelsize' : 15 })* before producing a visualization.

[8]:
```
sns.set(rc = {'figure.figsize' : ( 12, 5 ), 'axes.labelsize' : 10 })
```

[9]:
```
plt.
 ↪bar(gourd_df[(gourd_df["State"]=="ILLINOIS")]["Year"],gourd_df[(gourd_df["State"]=="ILLINOI
 ↪- ACRES PLANTED"], color="green", label="Acres Planted")
plt.
 ↪bar(gourd_df[(gourd_df["State"]=="ILLINOIS")]["Year"],gourd_df[(gourd_df["State"]=="ILLINOI
 ↪- ACRES HARVESTED"], color="orange", label="Acres Harvested")
plt.title("Acres of Pumpkin Crop Planted and Harvested in Illinois, 2000-2020")
plt.legend()
plt.show()
```



We're just looking for something that proves students can interpret their visualization. One or more solid observation(s) will work here. For example, full credit for:

Yes, from 2000-2012 there is a generally upward trend in the number of acres of pumpkins harvested.

1.0.4 Question 4

(3 points for code incl. visualization, 1 point for answer)

Suppose we're curious about the relationship between acres of squash planted, acres of squash harvested, and price received in $/cwd (the price per 100lbs of squash) across all states and years.

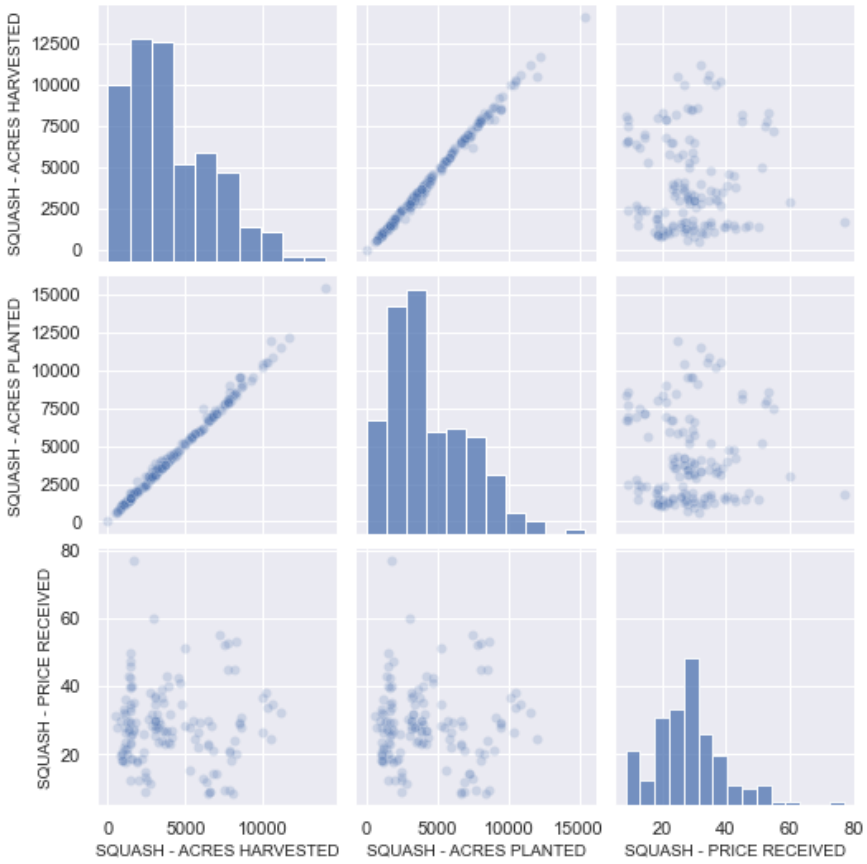
Produce a single visualization to demonstrate the relationship between the three variables. Briefly describe what you learn from your visualization. Does there seem to be a relationship between price and acres of pumpkins harvested?

Hint: if you're having trouble with title placement, look up and experiment with the suptitle() function.

[10]:
```
sns.set(rc = {'figure.figsize' : ( 20, 20 ), 'axes.labelsize' : 10 })
```

[11]:
```
pairplt = sns.pairplot(gourd_df[["SQUASH - ACRES HARVESTED","SQUASH - ACRES␣
↪PLANTED",\
                        "SQUASH - PRICE RECEIVED"]], \
             diag_kind='hist', plot_kws={'alpha': 0.2})
pairplt.fig.suptitle("Pairplot of Acres of Squash Harvested, Acres of Squash␣
↪Planted, and Price Received for Squash ($/cwd)",\
                     y=1.08)
plt.show()
```

Pairplot of Acres of Squash Harvested, Acres of Squash Planted, and Price Received for Squash ($/cwd)



We're just looking for something that proves students can interpret their visualization. One or more solid observation(s) will work here. For example, full credit for:

No, the plots of acres harvested and acres planted against price received look like somewhat random clouds of points. There is no clear relationship between the variables, and we may want to use alternative explanatory variables.

[ ]:

[ ]: