

#Question 1 Roller coasters are thrilling amusement park rides designed to make you squeal and scream! They take you up high, drop you to the ground quickly, and sometimes even spin you upside down before returning to a stop. Today you will be taking control back from the roller coasters and visualizing data covering international roller coaster rankings and roller coaster statistics. Roller coasters are often split into two main categories based on their construction material: wood or steel. Rankings for the best wood and steel roller coasters from the 2013 to 2018 Golden Ticket Awards are provided in 'Golden_Ticket_Award_Winners_Wood.csv' and 'Golden_Ticket_Award_Winners_Steel.csv', respectively. Load each csv into a DataFrame and inspect it to gain familiarity with the data. Open the hint for more information about each column of the datasets. #Question 2 Write a function that will plot the ranking of a given roller coaster over time as a line. Your function should take a roller coaster's name and a ranking DataFrame as arguments. Make sure to include informative labels that describe your visualization. Call your function with "El Toro" as the roller coaster name and the wood ranking DataFrame. What issue do you notice? Update your function with an additional argument to alleviate the problem, and retest your function.

```
In [95]: #Question 1
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as matplotlib

Wood_Rank=pd.read_csv(r'C:\Users\Apple\Desktop\新建文件夹 (10)\roller_coaster_starting')
Steel_Rank=pd.read_csv(r'C:\Users\Apple\Desktop\新建文件夹 (10)\roller_coaster_starting')
pd.set_option('display.width',1000)
print(Wood_Rank.head())
print(Steel_Rank.head())
```

	Rank	Name	Park	Location	Supplier
r	Year Built	Points	Year of Rank		
0	1	Boulder Dash	Lake Compounce	Bristol, Conn.	CC
I	2000	1333	2013		
1	2	El Toro	Six Flags Great Adventure	Jackson, N. J.	Intamin
n	2006	1302	2013		
2	3	Phoenix	Knoebels Amusement Resort	Elysburg, Pa.	Dinn/PTC-Schmect
k	1985	1088	2013		
3	4	The Voyage	Holiday World	Santa Claus, Ind.	Gravity Group
p	2006	1086	2013		
4	5	Thunderhead	Dollywood	Pigeon Forge, Tenn.	GCI
I	2004	923	2013		

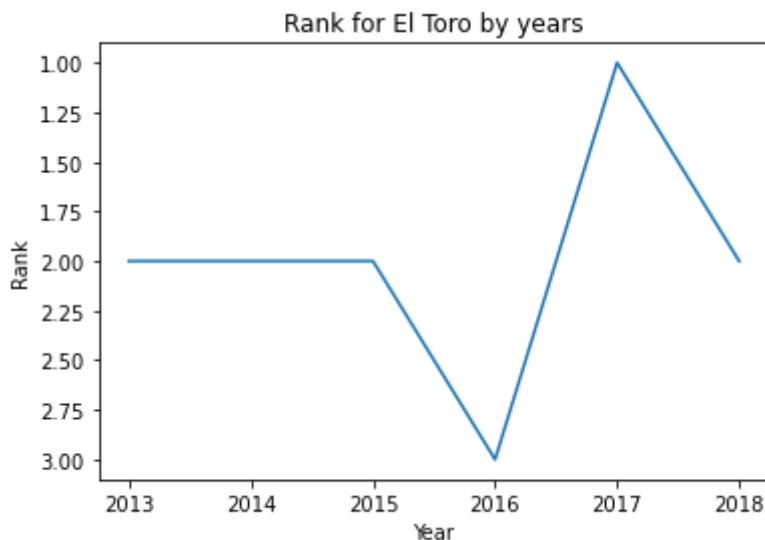
	Rank	Name	Park	Location	Supplier	Year
ear	Built	Points	Year of Rank			
0	1	Millennium Force	Cedar Point	Sandusky, Ohio	Intamin	
2000	1204		2013			
1	2	Bizarro	Six Flags New England	Agawam, Mass.	Intamin	
2000	1011		2013			
2	3	Expedition GeForce	Holiday Park	Hassloch, Germany	Intamin	
2001	598		2013			
3	4	Nitro	Six Flags Great Adventure	Jackson, N. J.	B&M	
2001	596		2013			
4	5	Apollo's Chariot	Busch Gardens Williamsburg	Williamsburg, Va.	B&M	
1999	542		2013			

Question 3

Write a function that will plot the ranking of two given roller coasters over time as lines. Your function should take both roller coasters' names and a ranking DataFrame as arguments. Make sure to include informative labels that describe your visualization.

Call your function with "El Toro" as one roller coaster name, "Boulder Dash" as the other roller coaster name, and the wood ranking DataFrame. What issue do you notice? Update your function with two additional arguments to alleviate the problem, and retest your function.

```
In [96]: #Question 3
ax=plt.subplot()
def rank_by_time (name,ranking_df):
    df_for_name=ranking_df[(ranking_df['Name']==name) & (ranking_df['Park']=='Six Fla
plt.plot(range(len(df_for_name)),df_for_name['Rank'])
ax.set_xticks(range(len(df_for_name)))
ax.set_xticklabels(df_for_name['Year of Rank'])
plt.title('Rank for '+name+' by years')
plt.xlabel('Year')
plt.ylabel('Rank')
plt.gca().invert_yaxis()
plt.show()
plt.clf()
rank_by_time ('El Toro',Wood_Rank)
```



<Figure size 432x288 with 0 Axes>

Question 4

Write a function that will plot the ranking of two given roller coasters over time as lines. Your function should take both roller coasters' names and a ranking DataFrame as arguments. Make sure to include informative labels that describe your visualization.

Call your function with "El Toro" as one roller coaster name, "Boulder Dash" as the other roller coaster name, and the wood ranking DataFrame. What issue do you notice? Update your function with two additional arguments to alleviate the problem, and retest your function.

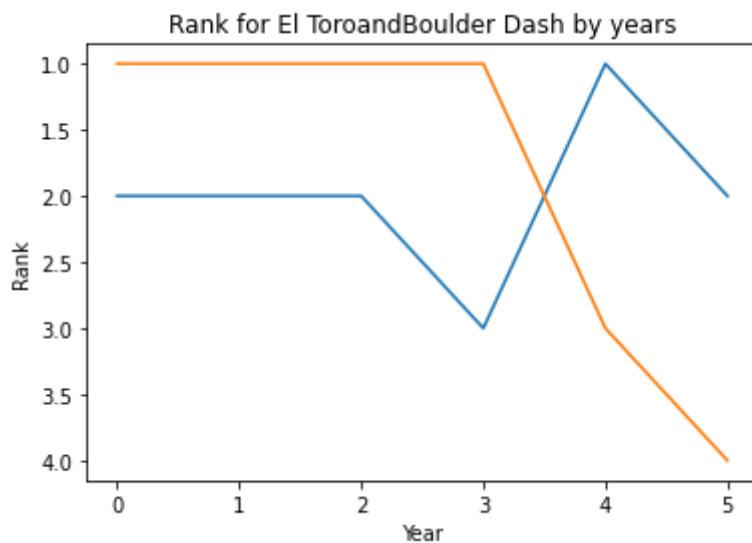
```
In [97]: #Question 4
def two_rank_by_time (name1,name2,ranking_df):
    df_for_name1=ranking_df[(ranking_df['Name']==name1) & (ranking_df['Park']=='Six Fla
    df_for_name2=ranking_df[(ranking_df['Name']==name2) & (ranking_df['Park']=='Six Fla
    df_for_name2.sort_values(by='Year of Rank')

    plt.plot(range(len(df_for_name1)),df_for_name1['Rank'])
    plt.plot(range(len(df_for_name2)),df_for_name2['Rank'])

    if len(df_for_name1)>=len(df_for_name2):
        ax.set_xticks(range(len(df_for_name1)))
        ax.set_xticklabels(df_for_name1['Year of Rank'])
    else:
        ax.set_xticks(range(len(df_for_name2)))
        ax.set_xticklabels(df_for_name2['Year of Rank'])
    plt.title('Rank for '+name1+' and '+name2+' by years')
```

```
plt.xlabel('Year')
plt.ylabel('Rank')
plt.gca().invert_yaxis()
plt.show()
plt.clf()

two_rank_by_time ('El Toro', 'Boulder Dash', Wood_Rank)
```



<Figure size 432x288 with 0 Axes>

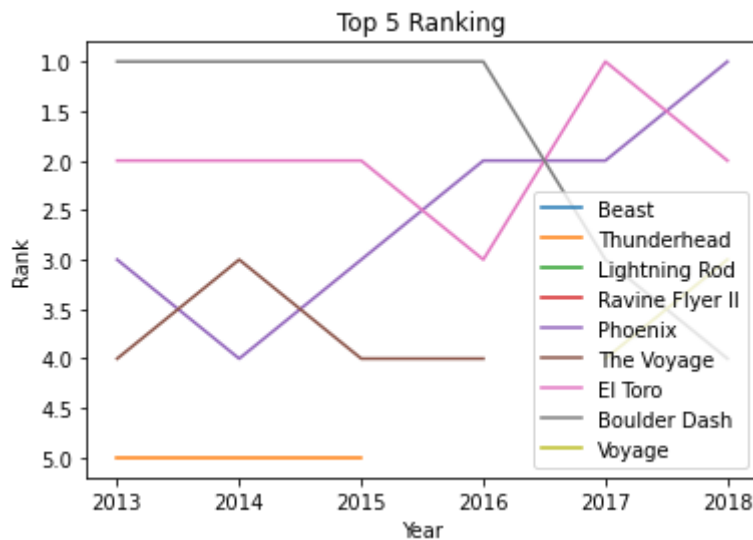
Question 5

Write a function that will plot the ranking of the top n ranked roller coasters over time as lines. Your function should take a number n and a ranking DataFrame as arguments. Make sure to include informative labels that describe your visualization.

For example, if n == 5, your function should plot a line for each roller coaster that has a rank of 5 or lower.

Call your function with a value for n and either the wood ranking or steel ranking DataFrame.

```
In [98]: #Question 5
def top_rank_by_time (n, rankings_df):
    top_n_rankings = rankings_df[rankings_df['Rank'] <= n]
    for coaster in set(top_n_rankings['Name']):
        coaster_rankings = top_n_rankings[top_n_rankings['Name'] == coaster]
        plt.plot(coaster_rankings['Year of Rank'], coaster_rankings['Rank'], label=coaster)
    plt.legend(loc=4)
    plt.xlabel('Year')
    plt.ylabel('Rank')
    plt.title('Top ' + str(n) + ' Ranking')
    plt.gca().invert_yaxis() #逆转y轴的顺序
    plt.show()
    plt.clf()
    top_rank_by_time (5, Wood_Rank)
```



<Figure size 432x288 with 0 Axes>

Question 6

Now that you've visualized rankings over time, let's dive into the actual statistics of roller coasters themselves. Captain Coaster is a popular site for recording roller coaster information. Data on all roller coasters documented on Captain Coaster has been accessed through its API and stored in roller_coasters.csv. Load the data from the csv into a DataFrame and inspect it to gain familiarity with the data.

Open the hint for more information about each column of the dataset.

Question 7

Write a function that plots a histogram of any numeric column of the roller coaster DataFrame. Your function should take a DataFrame and a column name for which a histogram should be constructed as arguments. Make sure to include informative labels that describe your visualization.

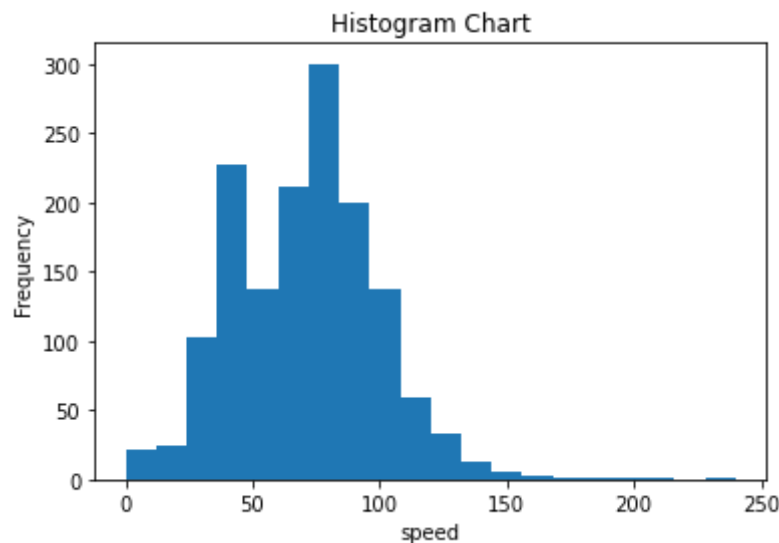
Call your function with the roller coaster DataFrame and one of the column names.

```
In [99]: #Question 6
Coaster=pd.read_csv(r'C:\Users\Apple\Desktop\新建文件夹 (10)\roller_coaster_starting\')
print(Coaster.head())

#Question 7
def question_7(df, column_name):
    target=df[column_name].dropna() #在画图之前记得先去除nan的值
    plt.hist(target,bins=20)
    plt.title('Histogram Chart')
    plt.xlabel(column_name)
    plt.ylabel('Frequency')
    plt.show()
    plt.clf()
    question_7(Coaster,'speed')
```

	name	material_type	seating_type	speed	height	length	num_inversions
manufacturer	park	status					
0	Goudurix	Steel	Sit Down	75.0	37.0	950.0	7.0

Vekoma	Parc Asterix	status.operating					
1	Dream catcher	Steel	Suspended	45.0	25.0	600.0	0.0
Vekoma	Bobbejaanland	status.operating					
2	Alucinakis	Steel	Sit Down	30.0	8.0	250.0	0.0
Zamperla	Terra Mítica	status.operating					
3	Anaconda	Wooden	Sit Down	85.0	35.0	1200.0	0.0
William J. Cobb	Walygator Parc	status.operating					
4	Azteka	Steel	Sit Down	55.0	17.0	500.0	0.0
Soquet	Le Pal	status.operating					



<Figure size 432x288 with 0 Axes>

Question 8

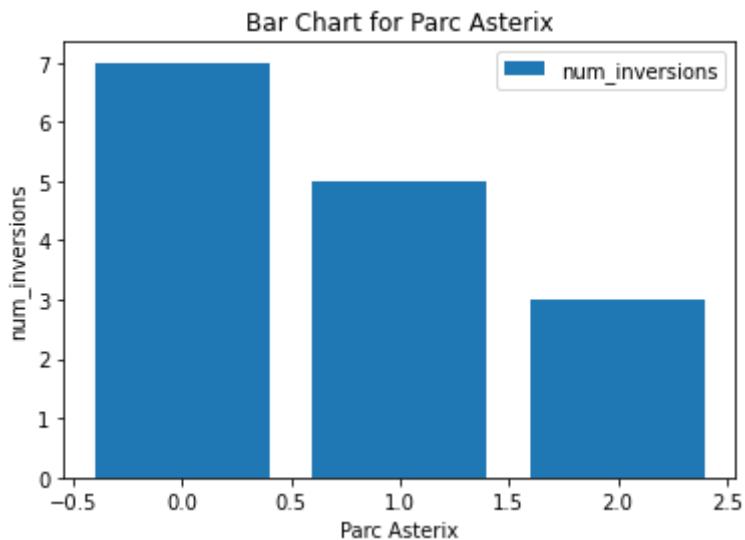
Write a function that creates a bar chart showing the number of inversions for each roller coaster at an amusement park. Your function should take the roller coaster DataFrame and an amusement park name as arguments. Make sure to include informative labels that describe your visualization.

Call your function with the roller coaster DataFrame and an amusement park name.

In [10...

```
#Question 8
def question_8(Coaster, park_name):
    row=Coaster[(Coaster['park']==park_name) & (Coaster['num_inversions']!=0)]
    plt.bar(range(len(row['name'])), row['num_inversions'], label='num_inversions')
    ax.set_xticks(range(len(row['name'])))
    ax.set_xticklabels(row['name'], fontsize=8, rotation = -30)
    plt.title('Bar Chart for '+park_name)
    plt.xlabel(park_name)
    plt.ylabel('num_inversions')
    plt.legend()
    plt.show()
    plt.clf()

question_8(Coaster, 'Parc Asterix')
```



<Figure size 432x288 with 0 Axes>

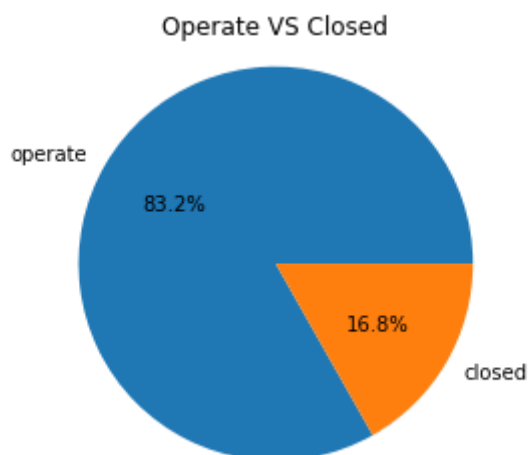
Question 9

Write a function that creates a pie chart that compares the number of operating roller coasters ('status.operating') to the number of closed roller coasters ('status.closed.definitely'). Your function should take the roller coaster DataFrame as an argument. Make sure to include informative labels that describe your visualization.

Call your function with the roller coaster DataFrame.

In [10...

```
#Question 9
def question_9_pie(Coaster):
    operating=len(Coaster[Coaster['status']=='status.operating'])
    closed=len(Coaster[Coaster['status']=='status.closed.definitely'])
    plt.pie([operating, closed], labels=['operate', 'closed'], autopct='%0.01f%%')
    plt.title('Operate VS Closed')
    plt.axis('equal')
    plt.show()
    plt.clf()
question_9_pie(Coaster)
```



<Figure size 432x288 with 0 Axes>

Question 10

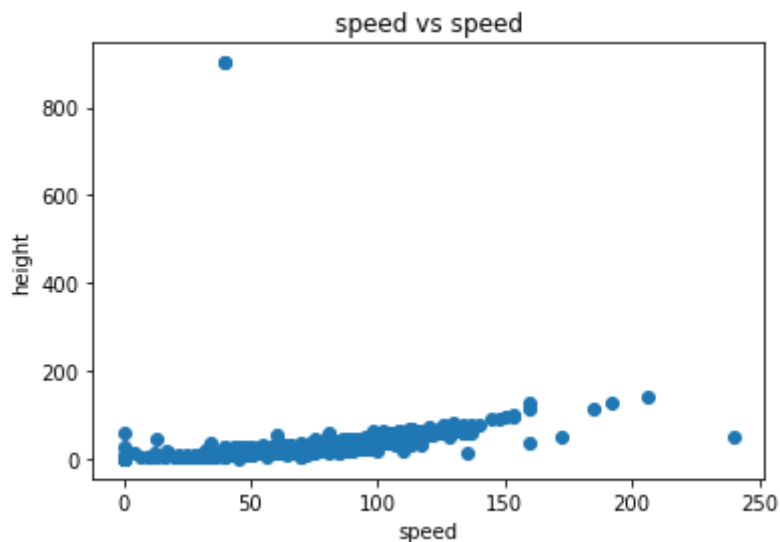
`.scatter()` is another useful function in matplotlib that you might not have seen before. `.scatter()` produces a scatter plot, which is similar to `.plot()` in that it plots points on a figure. `.scatter()`, however, does not connect the points with a line. This allows you to analyze the relationship between two variables. Find `.scatter()`'s documentation [here](#).

Write a function that creates a scatter plot of two numeric columns of the roller coaster DataFrame. Your function should take the roller coaster DataFrame and two-column names as arguments. Make sure to include informative labels that describe your visualization.

Call your function with the roller coaster DataFrame and two-column names.

In [10...

```
#Question 10
def question_10_scatter(Coaster, name1, name2):
    plt.scatter(Coaster[name1], Coaster[name2])
    plt.xlabel(name1)
    plt.ylabel(name2)
    plt.title('{} vs {}'.format(name1, name2))
    plt.show()
    plt.clf()
question_10_scatter(Coaster, 'speed', 'height')
```



<Figure size 432x288 with 0 Axes>

In []:

In []: