

Step 0: Import & Reading the data

```
In [22]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

plt.style.use('ggplot')
pd.set_option('display.max_columns', 200) # set số cột tối đa mà python có thể đọc được

In [110]: df = pd.read_csv('C:\\Users\\PC\\Desktop\\DATA ANALYST\\PORTFOLIO PROJECT\\exploratory d
coaster_db.csv')
```

Step 1: Data Understanding

```
In [8]: # Dataframe shape
# head and tail
# dtypes
# describe
```

```
In [111]: df.shape

Out[111]: (1087, 56)
```

```
In [23]: df.head(5)
```

Out[23]:

	coaster_name	Length	Speed	Location	Status	Opening date	Type	Manufacturer	Height restriction	Model	H
0	Switchback Railway	600 ft (180 m)	6 mph (9.7 km/h)	Coney Island	Removed	June 16, 1884	Wood	LaMarcus Adna Thompson	NaN	Lift Packed	(1
1	Flip Flap Railway	NaN	NaN	Sea Lion Park	Removed	1895	Wood	Lina Beecher	NaN	NaN	
2	Switchback Railway (Euclid Beach Park)	NaN	NaN	Cleveland, Ohio, United States	Closed	NaN	Other	NaN	NaN	NaN	
3	Loop the Loop (Coney Island)	NaN	NaN	Other	Removed	1901	Steel	Edwin Prescott	NaN	NaN	

In [24]: `df.columns`

Out[24]: Index(['coaster_name', 'Length', 'Speed', 'Location', 'Status', 'Opening date', 'Type', 'Manufacturer', 'Height restriction', 'Model', 'Height', 'Inversions', 'Lift/launch system', 'Cost', 'Trains', 'Park section', 'Duration', 'Capacity', 'G-force', 'Designer', 'Max vertical angle', 'Drop', 'Soft opening date', 'Fast Lane available', 'Replaced', 'Track layout', 'Fastrack available', 'Soft opening date.1', 'Closing date', 'Opened', 'Replaced by', 'Website', 'Flash Pass Available', 'Must transfer from wheelchair', 'Theme', 'Single rider line available', 'Restraint Style', 'Flash Pass available', 'Acceleration', 'Restrains', 'Name', 'year_introduced', 'latitude', 'longitude', 'Type_Main', 'opening_date_clean', 'speed1', 'speed2', 'speed1_value', 'speed1_unit', 'speed_mph', 'height_value', 'height_unit', 'height_ft', 'Inversions_clean', 'Gforce_clean'], dtype='object')

In [26]: `df.dtypes`

Out[26]:

coaster_name	object
Length	object
Speed	object
Location	object
Status	object
Opening date	object
Type	object
Manufacturer	object
Height restriction	object
Model	object
Height	object
Inversions	float64
Lift/launch system	object
Cost	object
Trains	object
Park section	object
Duration	object
Capacity	object
G-force	object
Designer	object
Max vertical angle	object
Drop	object
Soft opening date	object
Fast Lane available	object
Replaced	object
Track layout	object
Fastrack available	object
Soft opening date.1	object
Closing date	object
Opened	object
Replaced by	object
Website	object
Flash Pass Available	object
Must transfer from wheelchair	object
Theme	object
Single rider line available	object
Restraint Style	object
Flash Pass available	object
Acceleration	object
Restrains	object
Name	object

```

year_introduced      int64
latitude             float64
longitude            float64
Type_Main            object
opening_date_clean   object
speed1              object
speed2              object
speed1_value         float64
speed1_unit          object
speed_mph            float64
height_value         float64
height_unit          object
height_ft            float64
Inversions_clean     int64
Gforce_clean         float64
dtype: object

```

```
In [29]: df.describe()
```

```
Out[29]:
```

	Inversions	year_introduced	latitude	longitude	speed1_value	speed_mph	height_value	height_ft
count	932.000000	1087.000000	812.000000	812.000000	937.000000	937.000000	965.000000	171.000000
mean	1.547210	1994.986201	38.373484	-41.595373	53.850374	48.617289	89.575171	101.996491
std	2.114073	23.475248	15.516596	72.285227	23.385518	16.678031	136.246444	67.329092
min	0.000000	1884.000000	-48.261700	-123.035700	5.000000	5.000000	4.000000	13.100000
25%	0.000000	1989.000000	35.031050	-84.552200	40.000000	37.300000	44.000000	51.800000
50%	0.000000	2000.000000	40.289800	-76.653600	50.000000	49.700000	79.000000	91.200000
75%	3.000000	2010.000000	44.799600	2.778100	63.000000	58.000000	113.000000	131.200000
max	14.000000	2022.000000	63.230900	153.426500	240.000000	149.100000	3937.000000	377.300000

Step 2: Data Preparation

```
In [ ]: # Dropping irrelevant columns and rows
        # Identifying duplicated columns
        # Renaming Columns
        # Feature Creation
```

```
In [42]: df_new = df[['coaster_name',
                    #'Length', 'Speed',
                    #'Location', 'Status',
                    # 'Opening date','Type',
                    #'Manufacturer',
                    #'Height restriction', 'Model', 'Height',
                    # 'Inversions', 'Lift/launch system', 'Cost', 'Trains', 'Park section',
                    # 'Duration', 'Capacity', 'G-force', 'Designer', 'Max vertical angle',
                    # 'Drop', 'Soft opening date', 'Fast Lane available', 'Replaced',
                    # 'Track layout', 'Fastrack available', 'Soft opening date.1',
                    #'Closing date', 'Opened', 'Replaced by', 'Website',
                    # 'Flash Pass Available', 'Must transfer from wheelchair', 'Theme',
                    # 'Single rider line available', 'Restraint Style',
                    # 'Flash Pass available', 'Acceleration', 'Restrains', 'Name',
                    'year_introduced', 'latitude', 'longitude', 'Type_Main',
                    'opening_date_clean',
                    # 'speed1', 'speed2', 'speed1_value', 'speed1_unit',
                    'speed_mph',
                    # 'height_value', 'height_unit',
```

```
'height_ft',  
'Inversions_clean', 'Gforce_clean']]).copy()
```

```
In [50]: df_new['opening_date_clean'] = pd.to_datetime(df_new['opening_date_clean'])  
df_new['opening_date_clean']
```

```
Out[50]: 0      1884-06-16  
1      1895-01-01  
2           NaT  
3      1901-01-01  
4      1901-01-01  
      ...  
1082          NaT  
1083    2022-01-01  
1084    2016-06-16  
1085          NaT  
1086    2022-01-01  
Name: opening_date_clean, Length: 1087, dtype: datetime64[ns]
```

```
In [100]: df_new = df_new.rename(columns={  
    'coaster_name': 'Coaster_name',  
    'year_introduced' : 'Year_introduced',  
    'latitude' : 'Latitude',  
    'longitude' : 'Longitude',  
    'Type_Main' : 'Type_main',  
    'opening_date_clean' : 'Opening_date',  
    'speed_mph' : 'Speed_mph',  
    'height_ft' : 'Height_ft',  
    'Inversions_clean' : 'Inversions',  
    'Gforce_clean' : 'Gforce'  
})  
df_new.head(5)
```

```
Out[100]:
```

	Coaster_name	Location	Status	Manufacturer	Year_introduced	Latitude	Longitude	Type_main	Opening_
0	Switchback Railway	Coney Island	Removed	LaMarcus Adna Thompson	1884	40.5740	-73.9780	Wood	
1	Flip Flap Railway	Sea Lion Park	Removed	Lina Beecher	1895	40.5780	-73.9790	Wood	
2	Switchback Railway (Euclid Beach Park)	Cleveland, Ohio, United States	Closed	NaN	1896	41.5800	-81.5700	Other	
3	Loop the Loop (Coney Island)	Other	Removed	Edwin Prescott	1901	40.5745	-73.9780	Steel	
4	Loop the Loop (Young's Pier)	Other	Removed	Edwin Prescott	1901	39.3538	-74.4342	Steel	

```
In [93]: df_new.isna().sum()
```

```
Out[93]: Coaster_name      0  
Location      0  
Status      213  
Manufacturer    59  
Year_introduced    0  
Latitude      275  
Longitude      275  
Type_main      0  
Opening_date_clean    250  
Speed_mph      150
```

```
Height_ft      916
Inversions      0
Gforce         725
dtype: int64
```

```
In [94]: df_new.loc[df_new.duplicated()]
```

```
Out[94]: Coaster_name Location Status Manufacturer Year_introduced Latitude Longitude Type_main Opening_date_
```

```
In [119]: df_new.loc[df_new.duplicated(subset=['Coaster_name'])]
```

	Coaster_name	Location	Status	Manufacturer	Year_introduced	Latitude	Longitude	Type_main	C
43	Crystal Beach Cyclone	Crystal Beach Park	Removed	Traver Engineering	1927	42.8617	-79.0598	Wood	
60	Derby Racer	Revere Beach	Removed	Fred W. Pearce	1937	42.4200	-70.9860	Wood	
61	Blue Streak (Conneaut Lake)	Conneaut Lake Park	Closed	NaN	1938	41.6349	-80.3180	Wood	
167	Big Thunder Mountain Railroad	Other	NaN	Arrow Development (California and Florida)Dyna...	1980	NaN	NaN	Steel	
237	Thunder Run (Canada's Wonderland)	Canada's Wonderland	Operating	Mack Rides	1986	43.8427	-79.5423	Steel	
...
1063	Lil' Devil Coaster	Six Flags Great Adventure	Operating	Zamperla	2021	40.1343	-74.4434	Steel	
1064	Little Dipper (Conneaut Lake Park)	Conneaut Lake Park	Operating	Allan Herschell Company	2021	41.6343	-80.3165	Steel	
1080	Iron Gwazi	Busch Gardens Tampa Bay	Under construction	Rocky Mountain Construction	2022	28.0339	-82.4231	Steel	
1082	American Dreier Looping	Other	NaN	Anton Schwarzkopf	2022	NaN	NaN	Steel	
1084	Tron Lightcycle Power Run	Other	NaN	Vekoma	2022	NaN	NaN	Steel	

97 rows × 13 columns

```
In [95]: # checking an example of duplicate
df_new.query('Coaster_name == "Crystal Beach Cyclone"')
# we have some faults in Year_introduced' columns
# -> so we will check the data by 'Coaster_name', 'Location', 'Opening_date_clean' to fi
```

```
Out[95]: Coaster_name Location Status Manufacturer Year_introduced Latitude Longitude Type_main Opening_
```

39	Crystal Beach Cyclone	Crystal Beach	Removed	Traver Engineering	1926	42.8617	-79.0598	Wood	
----	-----------------------	---------------	---------	--------------------	------	---------	----------	------	--

		Park							
43	Crystal Beach Cyclone	Crystal Beach Park	Removed	Traver Engineering	1927	42.8617	-79.0598	Wood	

```
In [121]: df_new = df_new.loc[~df_new.duplicated(subset = ['Coaster_name', 'Location', 'Opening_da
df_new.shape
```

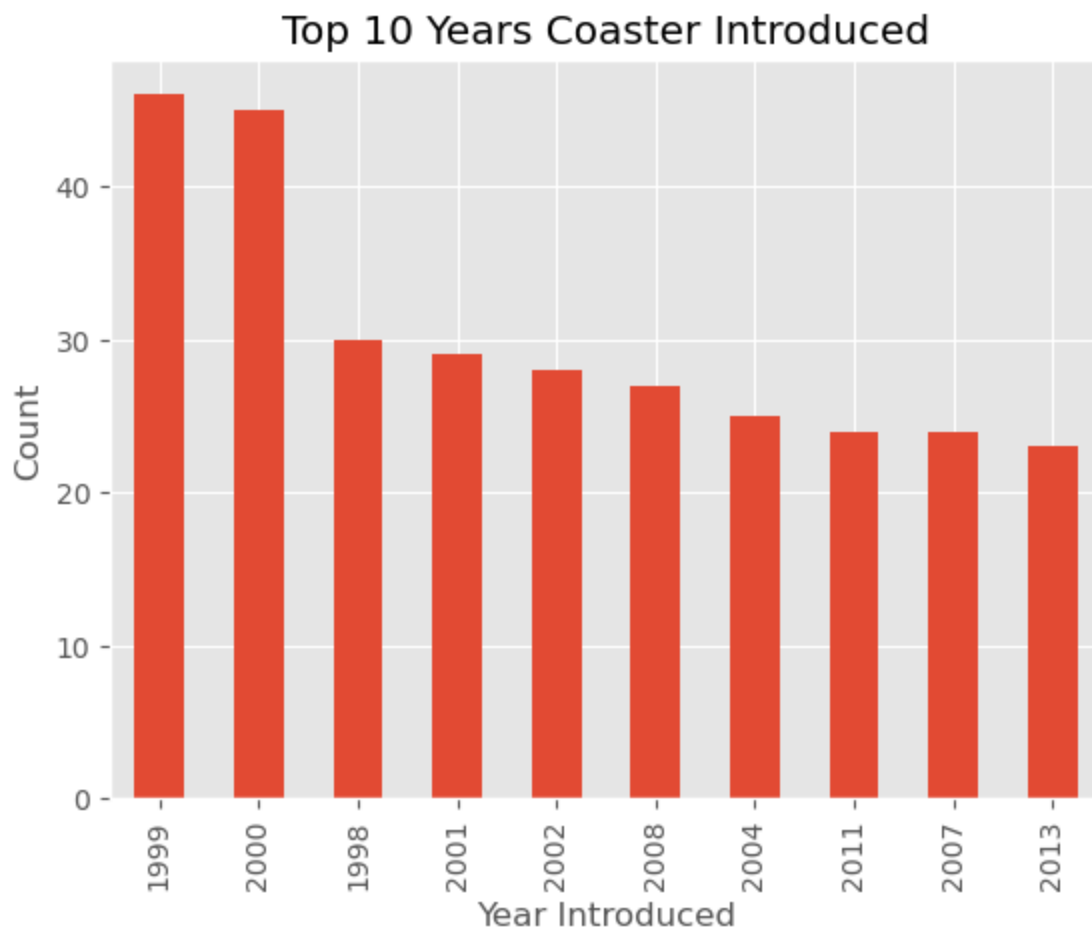
Out[121]: (990, 13)

STEP 3: Feature Understanding

```
In [ ]: # Plotting Feature Distributions
# Histogram
# KDE
# Boxplot
```

```
In [130]: year_intro = df_new['Year_introduced'].value_counts().head(10) \
.plot(kind = 'bar', title = 'Top 10 Years Coaster Introduced')
year_intro.set_xlabel('Year Introduced')
year_intro.set_ylabel('Count')
```

Out[130]: Text(0, 0.5, 'Count')

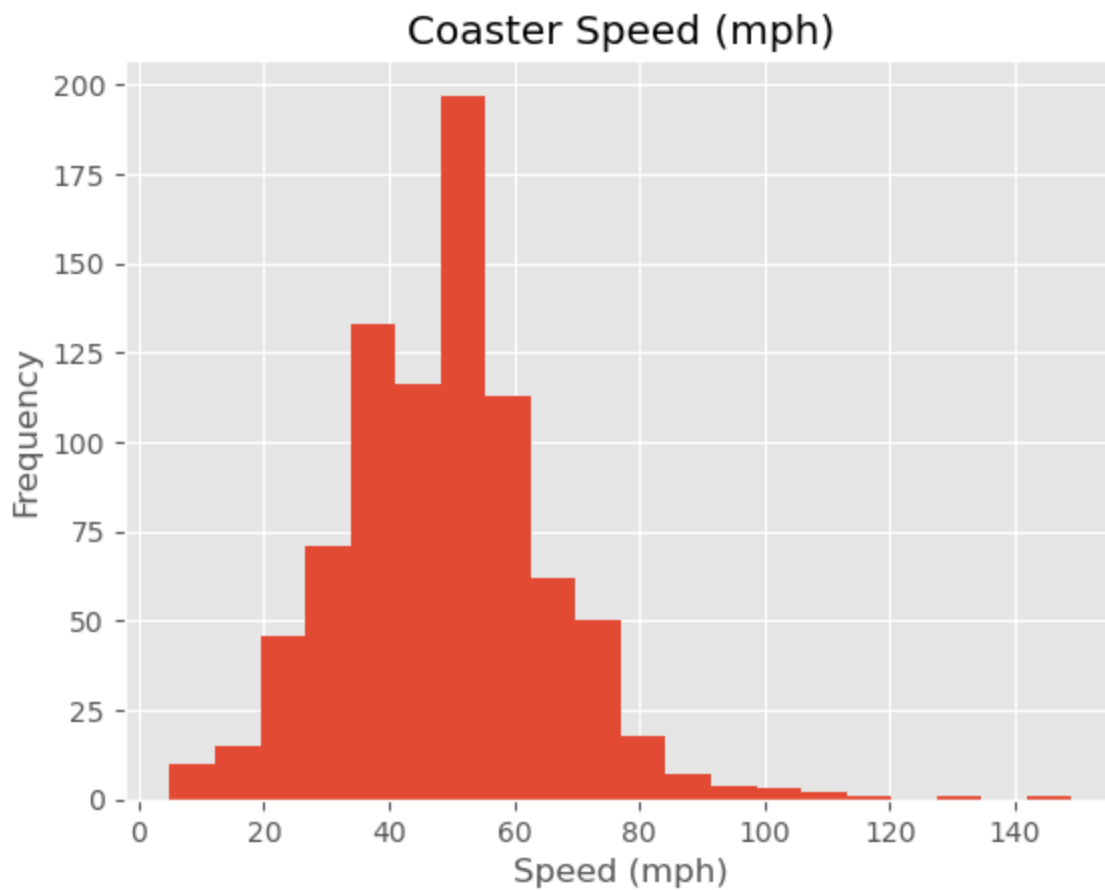


The largest number of roller coasters introduced is in 1999.

```
In [143]: Speed = df_new['Speed_mph'].plot(kind = 'hist', bins = 20, title = 'Coaster Speed (mph) '
Speed.set_xlabel('Speed (mph) ')
```

Text(0.5, 0, 'Speed (mph)')

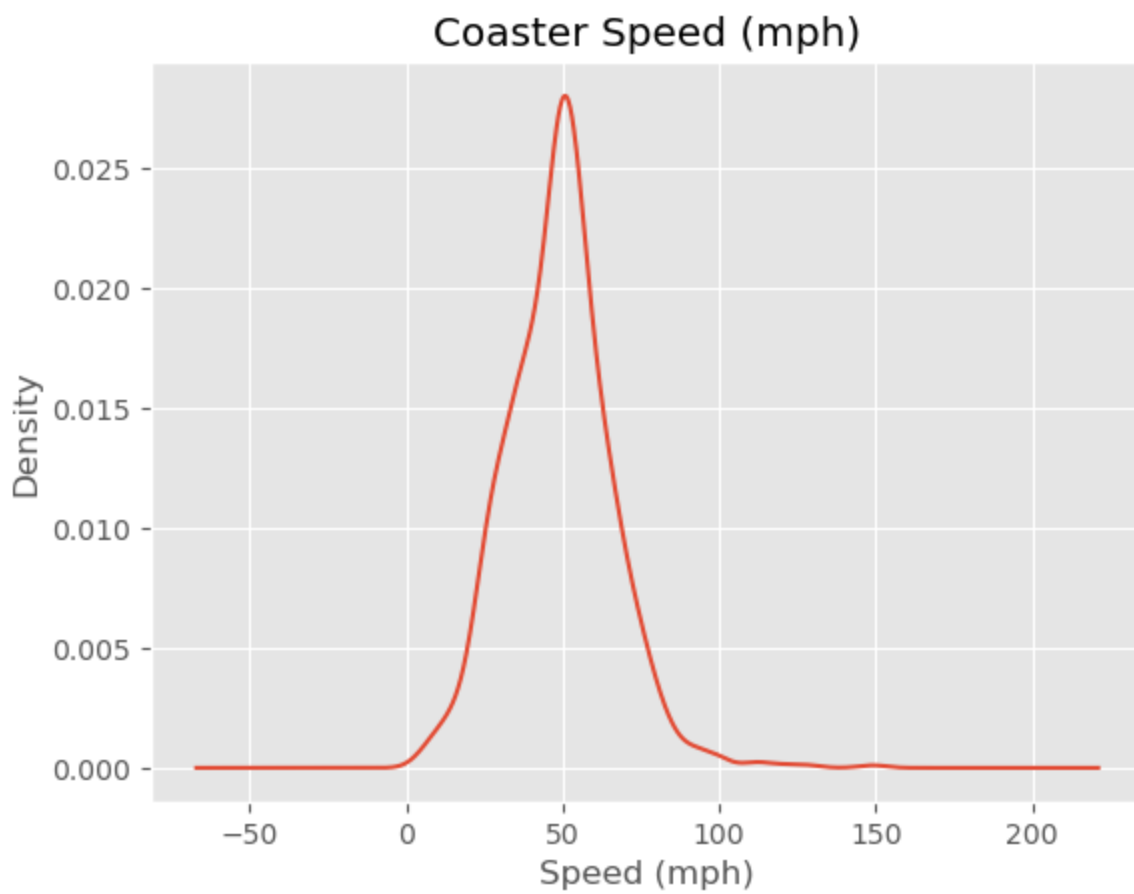
Out[143]:



The majority of roller coasters have the speed from 50mph to 60 mph.

```
In [146... Speed = df_new['Speed_mph'].plot(kind = 'kde', title = 'Coaster Speed (mph)')
Speed.set_xlabel('Speed (mph)')
```

Out[146]: Text(0.5, 0, 'Speed (mph)')

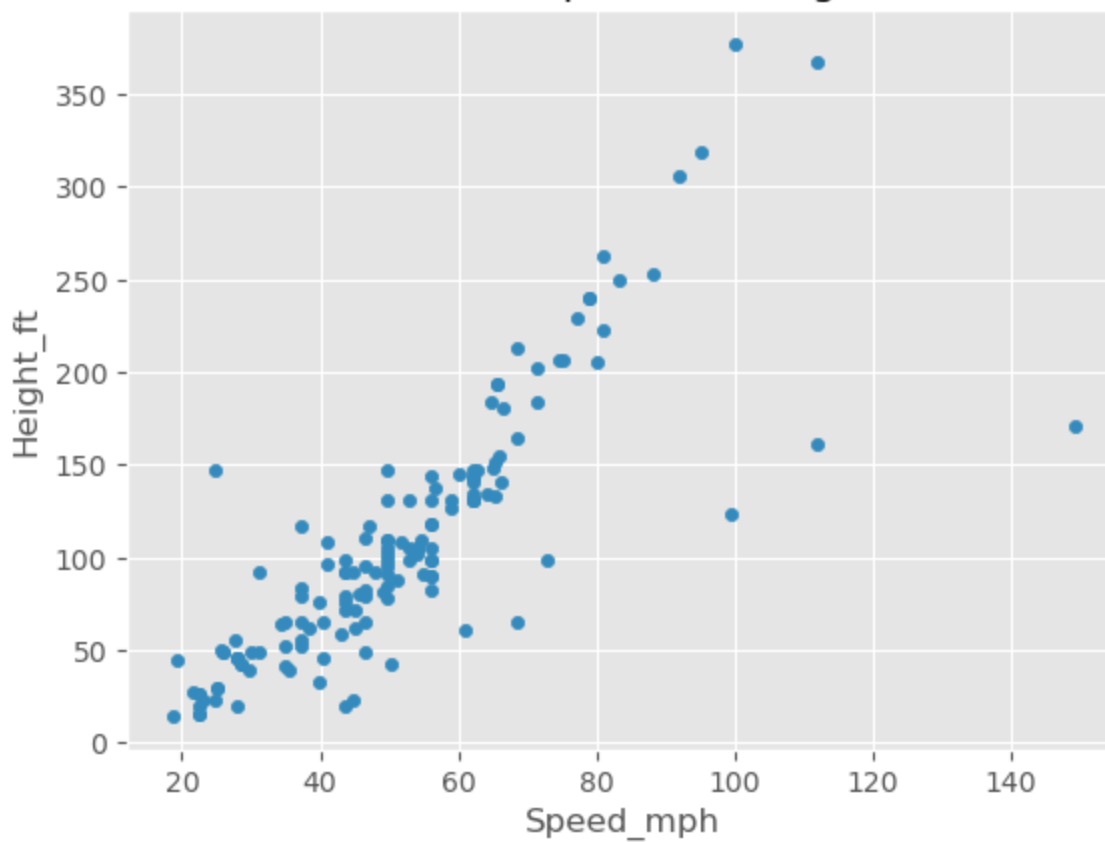


STEP 4: Feature Relationships

```
In [ ]: # Scatterplot  
        # Heatmap Correlation  
        # Pairplot  
        # Groupby comparisons
```

```
In [148... speed_vs_height = df_new.plot(kind = 'scatter', x='Speed_mph', y='Height_ft', title = 'C  
plt.show()
```

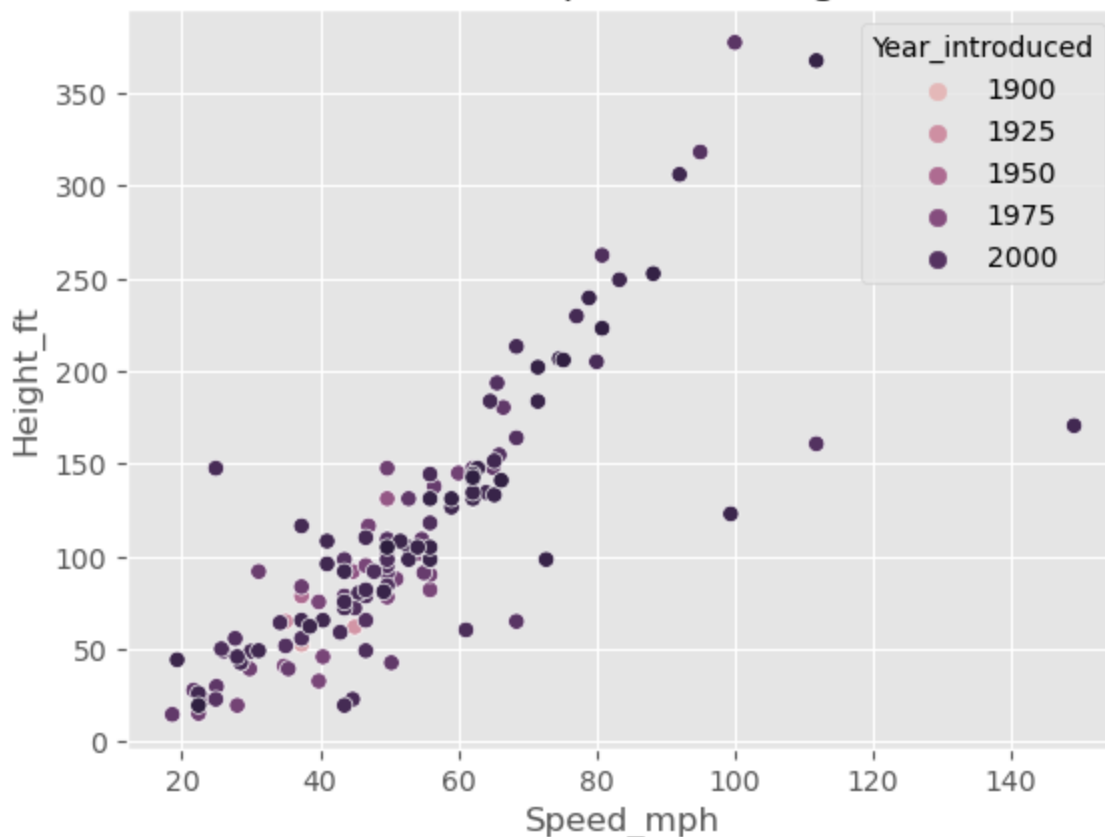

Coaster Speed vs. Height



As we can see, we have a trend that the higher the roller coaster is, the faster the speed is.

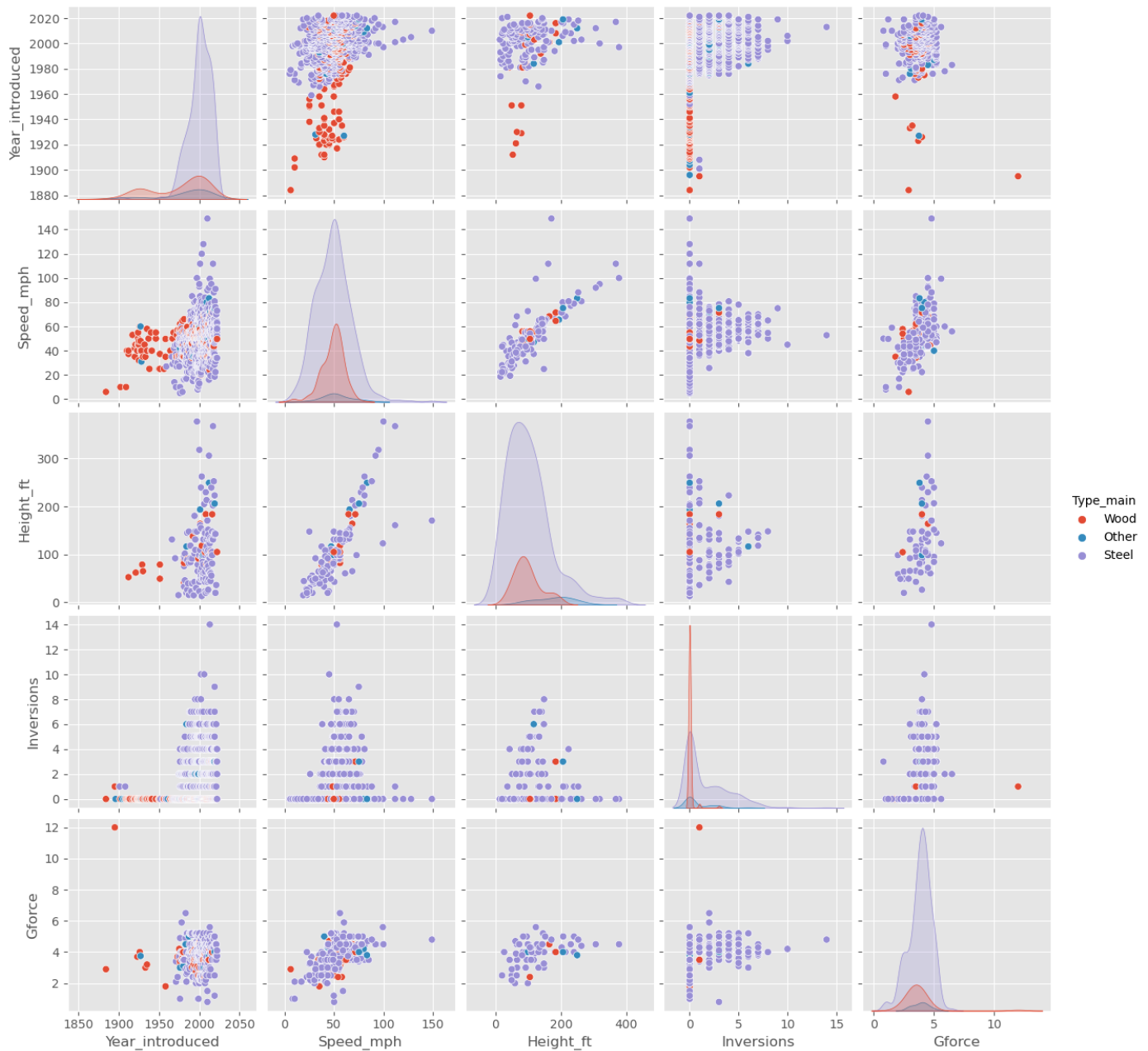
```
In [155... speed_vs_height = sns.scatterplot(x='Speed_mph', y='Height_ft', hue='Year_introduced', d
speed_vs_height.set_title('Coaster Speed vs. Height')
plt.show()
```

Coaster Speed vs. Height



```
In [157... sns.pairplot(df_new,
              vars=['Year_introduced', 'Speed_mph',
                   'Height_ft', 'Inversions', 'Gforce'],
              hue='Type_main')
plt.show()
```

C:\Users\PC\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)



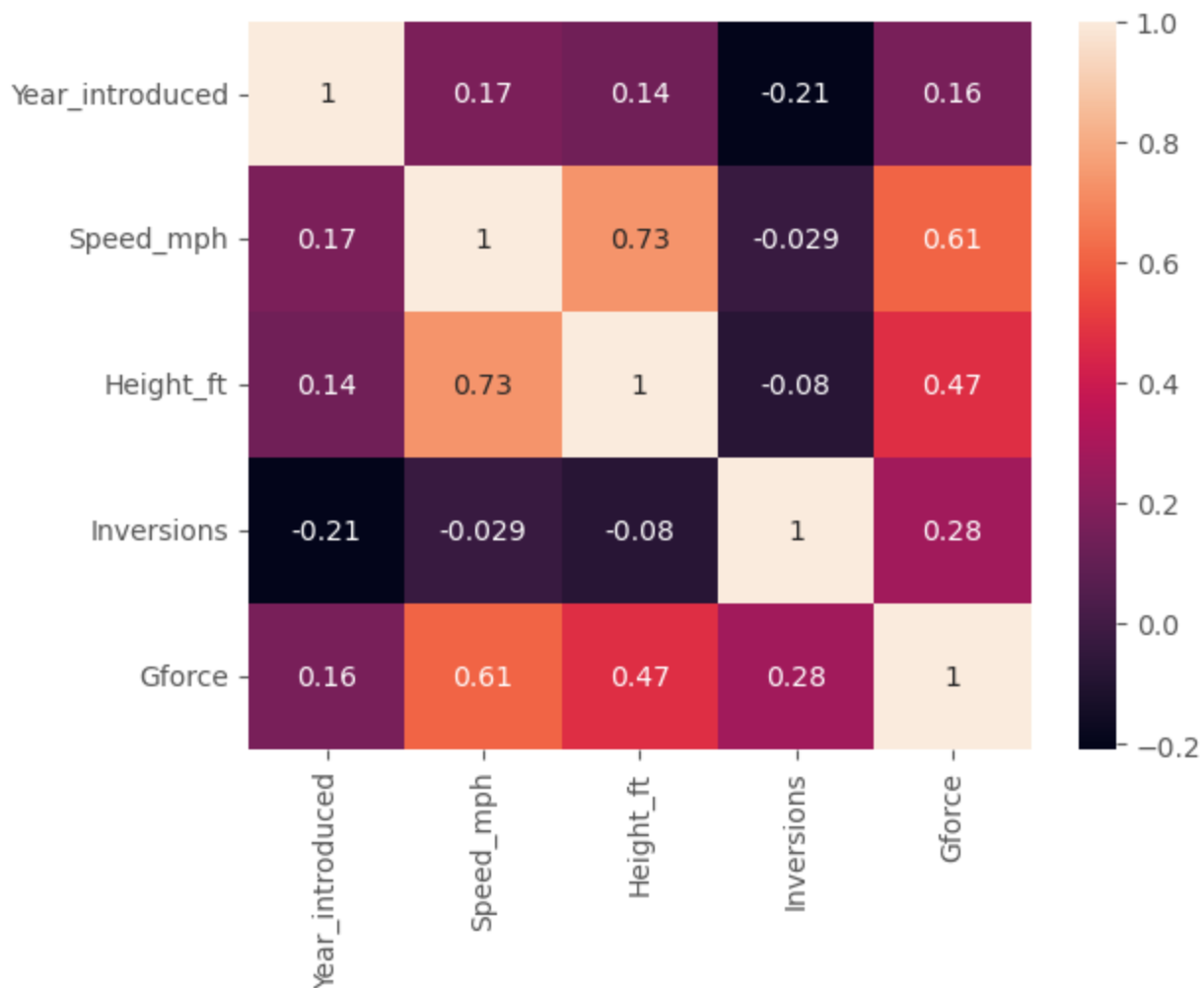
```
In [162... df_new_corr = df_new[['Year_introduced', 'Speed_mph',
                        'Height_ft', 'Inversions', 'Gforce']].dropna().corr()
df_new_corr
```

Out[162]:

	Year_introduced	Speed_mph	Height_ft	Inversions	Gforce
Year_introduced	1.000000	0.171978	0.135413	-0.209887	0.160247
Speed_mph	0.171978	1.000000	0.733999	-0.028705	0.607383
Height_ft	0.135413	0.733999	1.000000	-0.079736	0.466482
Inversions	-0.209887	-0.028705	-0.079736	1.000000	0.275991
Gforce	0.160247	0.607383	0.466482	0.275991	1.000000

```
In [164]: sns.heatmap(df_new_corr, annot=True)
```

```
Out[164]: <Axes: >
```



```
In [166]: # What are the location with fastest roller coaster (minimum of 10)?
ax = df_new.query('Location != "Other"') \
    .groupby('Location')['Speed_mph'] \
    .agg(['mean', 'count']) \
    .query('count >= 10') \
    .sort_values('mean')['mean'] \
    .plot(kind='barh', figsize=(12, 5), title='Average Coast Speed by Location')
ax.set_xlabel('Average Coaster Speed')
plt.show()
```

