

# Visualizing Data in Python

When working with a new dataset, one of the most useful things to do is to begin to visualize the data. By using **tables**, **histograms**, **boxplots**, **scatter plots** and other visual tools, we can get a better idea of what the data may be trying to tell us, and we can gain insights into the data that we may have not discovered otherwise.

In this notebook will use the [Seaborn](https://seaborn.pydata.org/) data processing library, which is a higher-level interface to **Matplotlib** that can be used to simplify many visualization tasks

The **Seaborn** provides visualisations tools that will allow to explore data from a graphical perspective.

## Acknowledgments

- Data from <https://www.coursera.org/> from the course "Understanding and Visualizing Data with Python" by University of Michigan

## ✓ Importing libraries

```
1 # Import the packages that we will be using
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import seaborn as sns
```

## ✓ Importing data

```
1 # Define where you are running the code: colab or local
2 RunInColab = True # (False: no | True: yes)
3
4 # If running in colab:
5 if RunInColab:
6     # Mount your google drive in google colab
7     from google.colab import drive
8     drive.mount('/content/drive')
9
10    # Find location
11    #!pwd
12    #!ls
13    #!ls "/content/drive/My Drive/Colab Notebooks/MachineLearningWithPython/"
14
15    # Define path del proyecto
16    Ruta = "/content/drive/MyDrive/ITC/5toSem/semanaTecAn/"
17
18 else:
19    # Define path del proyecto
20    Ruta = ""
```


 Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_r

```
1 # Dataset url
2 url = "datasets/cartwheel/cartwheel.csv"
3
4 # Load the dataset
5 df = pd.read_csv(Ruta + url)
```

## ✓ Exploring the content of the data set

Get a general 'feel' of the data

```
1 df
2 # Age is the age of participant doing a cartwheel
3 # Gender = gender of participant
4 # GenderGroup = F:1, M:2.
5 # Glasses? Yes, they use glasses or no, they dont
6 # Height of the person in inches
7 # Wingspan in inches
8 # CWD = CartWheelDistance in inches
9 # Complete? Did they complete the cartwheel: Yes or not
10 # CompleteGroup: Yes:1, No:0
11 # Score = How good of a grade the cartwheel deserves
```



	ID	Age	Gender	GenderGroup	Glasses	GlassesGroup	Height	Wingspan	CWDistance	Complete	CompleteGroup	Score
0	1	56.0	F	1	Y	1	62.00	61.0	79	Y	1.0	
1	2	26.0	F	1	Y	1	62.00	60.0	70	Y	1.0	
2	3	33.0	F	1	Y	1	66.00	64.0	85	Y	1.0	
3	4	39.0	F	1	N	0	64.00	63.0	87	Y	1.0	
4	5	27.0	M	2	N	0	73.00	75.0	72	N	0.0	
5	6	24.0	M	2	N	0	75.00	71.0	81	N	0.0	
6	7	28.0	M	2	N	0	75.00	76.0	107	Y	1.0	
7	8	22.0	F	1	N	0	65.00	62.0	98	Y	1.0	
8	9	29.0	M	2	Y	1	74.00	73.0	106	N	0.0	
9	10	33.0	F	1	Y	1	63.00	60.0	65	Y	1.0	
10	11	30.0	M	2	Y	1	69.50	66.0	96	Y	1.0	
11	12	28.0	F	1	Y	1	62.75	58.0	79	Y	1.0	
12	13	25.0	F	1	Y	1	65.00	64.5	92	Y	1.0	
13	14	23.0	F	1	N	0	61.50	57.5	66	Y	1.0	
14	15	31.0	M	2	Y	1	73.00	74.0	72	Y	1.0	
15	16	26.0	M	2	Y	1	71.00	72.0	115	Y	1.0	
16	17	26.0	F	1	N	0	61.50	59.5	90	N	0.0	
17	18	27.0	M	2	N	0	66.00	66.0	74	Y	1.0	
18	19	23.0	M	2	Y	1	70.00	69.0	64	Y	1.0	
19	20	24.0	F	1	Y	1	68.00	66.0	85	Y	1.0	
20	21	23.0	M	2	Y	1	69.00	67.0	66	N	0.0	
21	22	29.0	M	2	N	0	71.00	70.0	101	Y	1.0	
22	23	25.0	M	2	N	0	70.00	68.0	82	Y	1.0	
23	24	26.0	M	2	N	0	69.00	71.0	63	Y	1.0	
24	25	23.0	F	1	Y	1	65.00	63.0	67	N	0.0	
25	26	28.0	M	2	N	0	75.00	76.0	111	Y	1.0	
26	27	24.0	M	2	N	0	78.40	71.0	92	Y	1.0	
27	28	25.0	M	2	Y	1	76.00	73.0	107	Y	1.0	
28	29	32.0	F	1	Y	1	63.00	60.0	75	Y	1.0	
29	30	38.0	F	1	Y	1	61.50	61.0	78	Y	1.0	
30	31	27.0	F	1	Y	1	62.00	60.0	72	Y	1.0	
31	32	33.0	F	1	Y	1	65.30	64.0	91	Y	1.0	
32	33	38.0	F	1	N	0	64.00	63.0	86	Y	1.0	
33	34	27.0	M	2	N	0	77.00	75.0	100	Y	1.0	
34	35	24.0	F	1	N	0	67.80	62.0	98	Y	1.0	
35	36	27.0	M	2	N	0	68.00	66.0	74	Y	1.0	
36	37	25.0	F	1	Y	1	65.00	64.5	92	Y	1.0	
37	38	26.0	F	1	N	0	61.50	59.5	90	Y	1.0	
38	39	31.0	M	2	Y	1	73.00	74.0	72	Y	1.0	

39	40	30.0	M	2	Y	1	69.50	66.0	96	Y	1.0
40	41	23.0	F	1	N	0	70.40	71.0	66	Y	1.0
41	42	26.0	M	2	Y	1	73.50	72.0	115	Y	1.0
42	43	28.0	F	1	Y	1	72.50	72.0	81	Y	1.0
43	44	26.0	F	1	Y	1	72.00	72.0	92	Y	1.0
44	45	30.0	F	1	Y	1	66.00	64.0	85	Y	1.0
45	46	39.0	F	1	N	0	64.00	63.0	87	Y	1.0
46	47	27.0	M	2	N	0	78.00	75.0	72	N	0.0
47	48	24.0	M	2	N	0	79.50	75.0	82	N	0.0
48	49	28.0	M	2	N	0	77.80	76.0	99	Y	1.0
49	50	30.0	F	1	N	0	74.60	NaN	71	Y	1.0
50	51	NaN	M	2	N	0	71.00	70.0	101	Y	NaN
51	52	27.0	M	2	N	0	NaN	71.5	103	Y	1.0

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

## Frequency tables

The `value_counts()` method can be used to determine the number of times that each distinct value of a variable occurs in a data set. In statistical terms, this is the "frequency distribution" of the variable. The `value_counts()` method produces a table with two columns. The first column contains all distinct observed values for the variable. The second column contains the number of times each of these values occurs. Note that the table returned by `value_counts()` is actually a **Pandas** data frame, so can be further processed using any Pandas methods for working with data frames.

```
1 df.CompleteGroup.value_counts()
```



```

count
CompleteGroup
1.0      43
0.0       8

```

```
dtype: int64
```

```

1 # Proportion of each distinct value of a variable occurs in a data set
2 x= df.CompleteGroup.value_counts()
3 x= 100*x/x.sum()
4 x
5 #84.313725 si completaron la CW
6 #15.686275 no

```



```

count
CompleteGroup
1.0    84.313725
0.0    15.686275

```

```
dtype: float64
```

Note that the `value_counts()` method excludes missing values. We confirm this below by adding up observations to your data frame with some missing values and then computing `value_counts()` and comparing this to the total number of rows in the data set, which is 28. This tells us that there are  $28 - (21+6) = 1$  missing values for this variable (other variables may have different numbers of missing values).

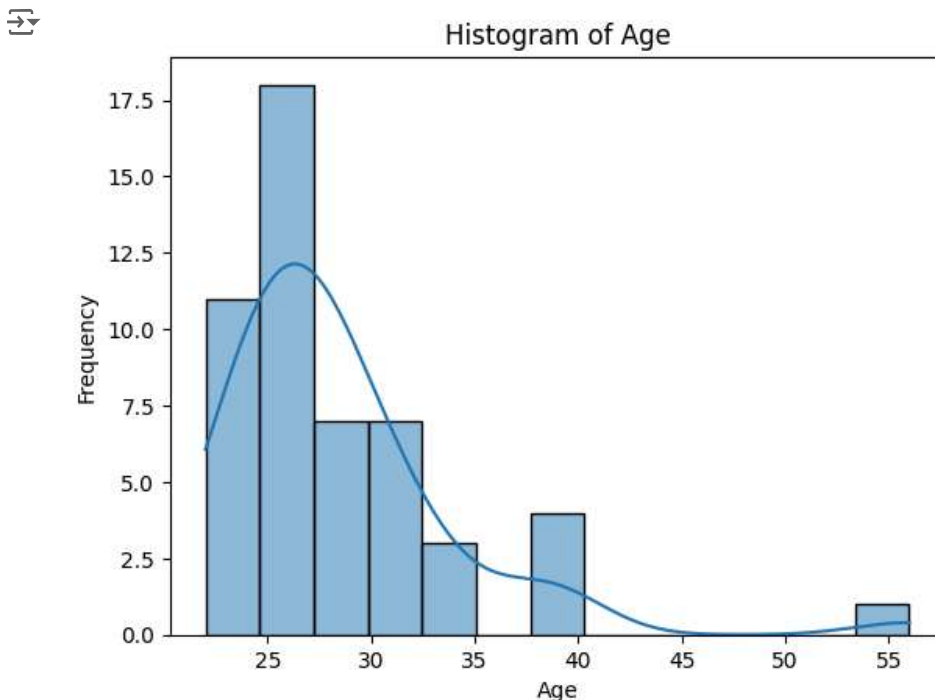
```
1 # Total number of observations
2 print("Total number of observations: " + str(df.shape[0]))
3
4 # Total number of null observations
5 print("Total number of null observations in Age: " + str(df.Age.isnull().sum()))
6 print("Total number of null observations in CompleteGroup: " + str(df.CompleteGroup.isnull().sum()))
7
8 # Total number of counts (excluding missing values)
9 print("Total number of counts in Age excluding missing values: " + str(df.Age.notnull().sum()))
```

```
➦ Total number of observations: 52
  Total number of null observations in Age: 1
  Total number of null observations in CompleteGroup: 1
  Total number of counts in Age excluding missing values: 51
```

## ✓ Histogram

It is often good to get a feel for the shape of the distribution of the data.

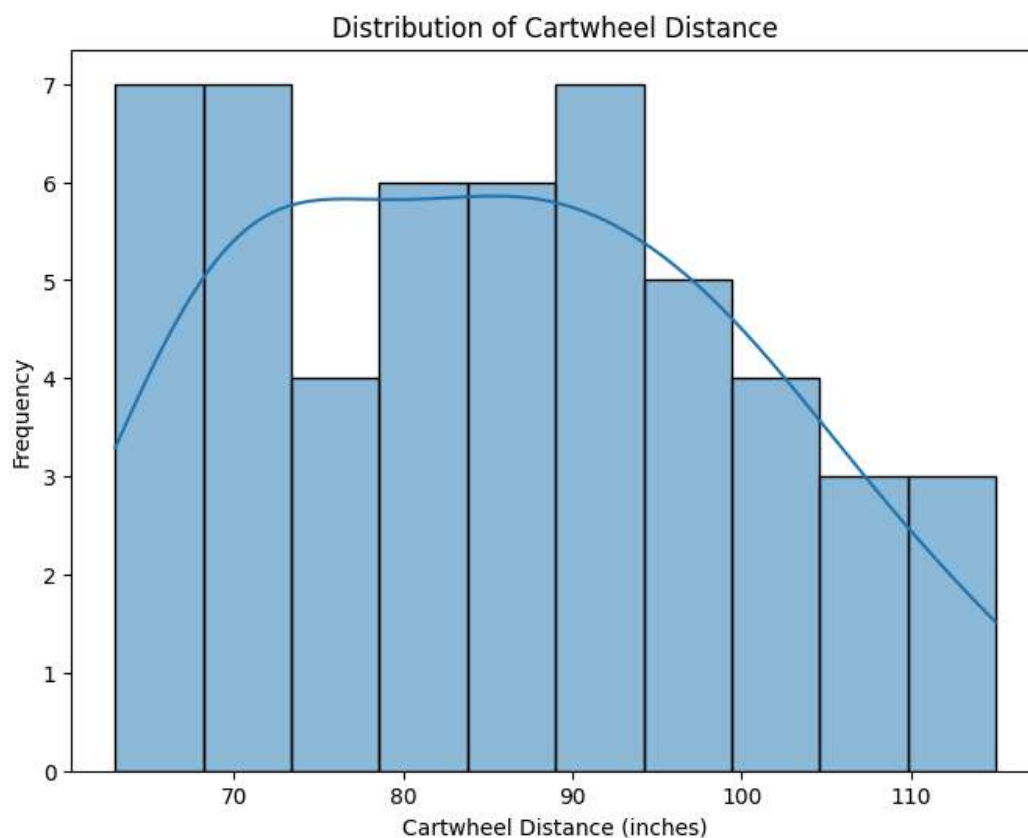
```
1 # Plot histogram of the total bill only
2 var2plot = df['Age']
3 sns.histplot(var2plot, kde=True) # kde=True adds a kernel density estimate
4 plt.title('Histogram of Age')
5 plt.xlabel('Age')
6 plt.ylabel('Frequency')
7 plt.show()
8
```



```

1 # Plot distribution of Cartwheel distance
2 plt.figure(figsize=(8, 6))
3 sns.histplot(df['CWDistance'].dropna(), kde=True, bins=10)
4 plt.title('Distribution of Cartwheel Distance')
5 plt.xlabel('Cartwheel Distance (inches)')
6 plt.ylabel('Frequency')
7 plt.show()

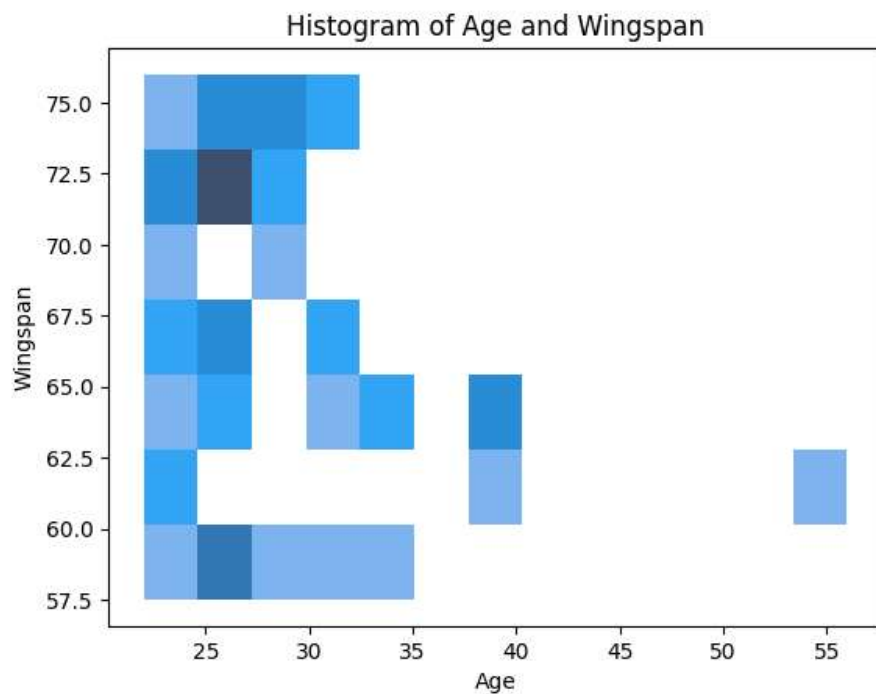
```



```

1 # Plot histogram of Age and Wingspan with Frequency
2 var1= df['Age']
3 var2= df['Wingspan']
4 sns.histplot(data=df, x=var1, y=var2)
5 plt.xlabel('Age')
6 plt.ylabel('Wingspan')
7 plt.title('Histogram of Age and Wingspan')
8 plt.show()

```

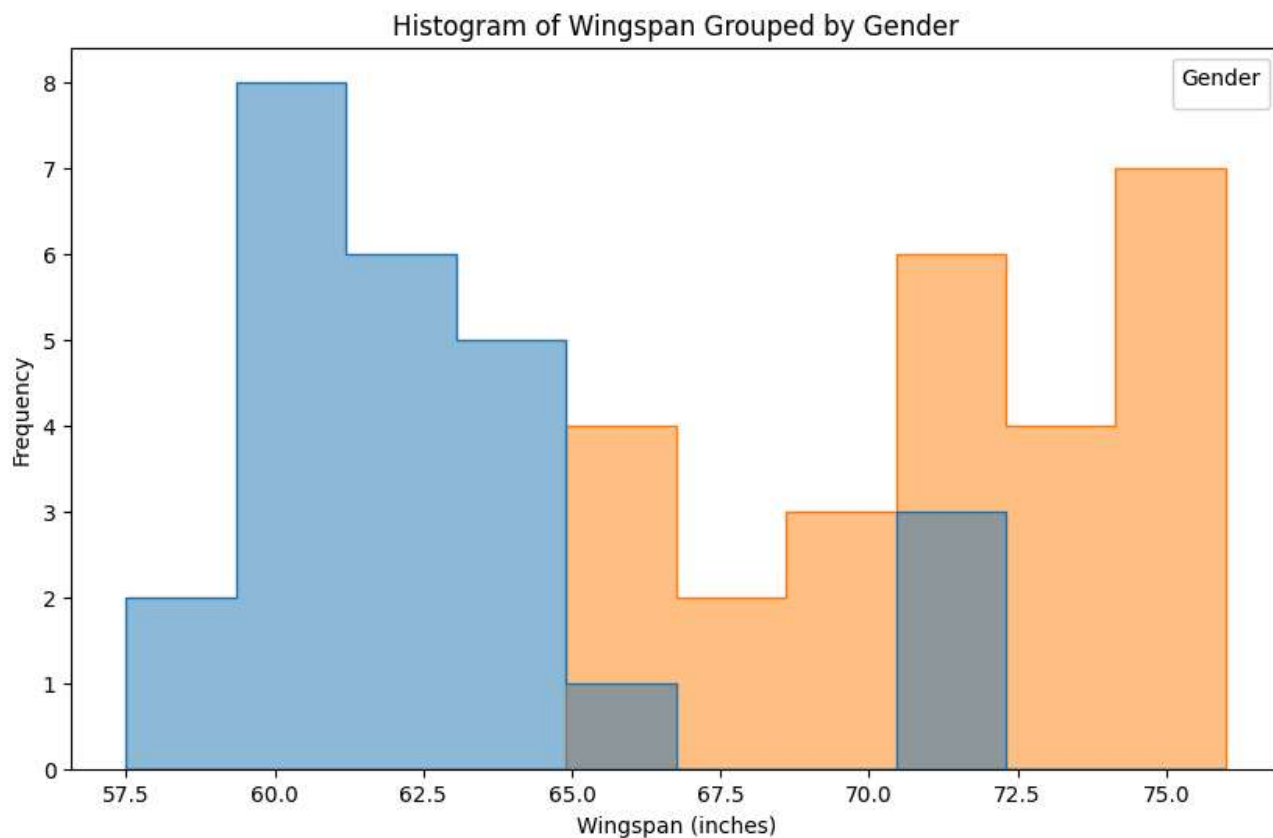


## ✓ Histograms plotted by groups

While looking at a single variable is interesting, it is often useful to see how a variable changes in response to another. Thus, we can create a histograms of one quantitative variable grouped by another categorical variables.

```
1 # Create histograms of the "Wingspan" grouped by "Gender"
2 plt.figure(figsize=(10, 6))
3 sns.histplot(data=df, x='Wingspan', hue='Gender', kde=False, bins=10, element='step', alpha=0.5)
4
5 # Add titles and labels
6 plt.title('Histogram of Wingspan Grouped by Gender')
7 plt.xlabel('Wingspan (inches)')
8 plt.ylabel('Frequency')
9 plt.legend(title='Gender') # Display the legend with the title 'Gender'
10
11 # Display the plot
12 plt.show()
```

⚠️ WARNING:matplotlib.legend.No artists with labels found to put in legend. Note that artists whose label start with



## ✓ Boxplots

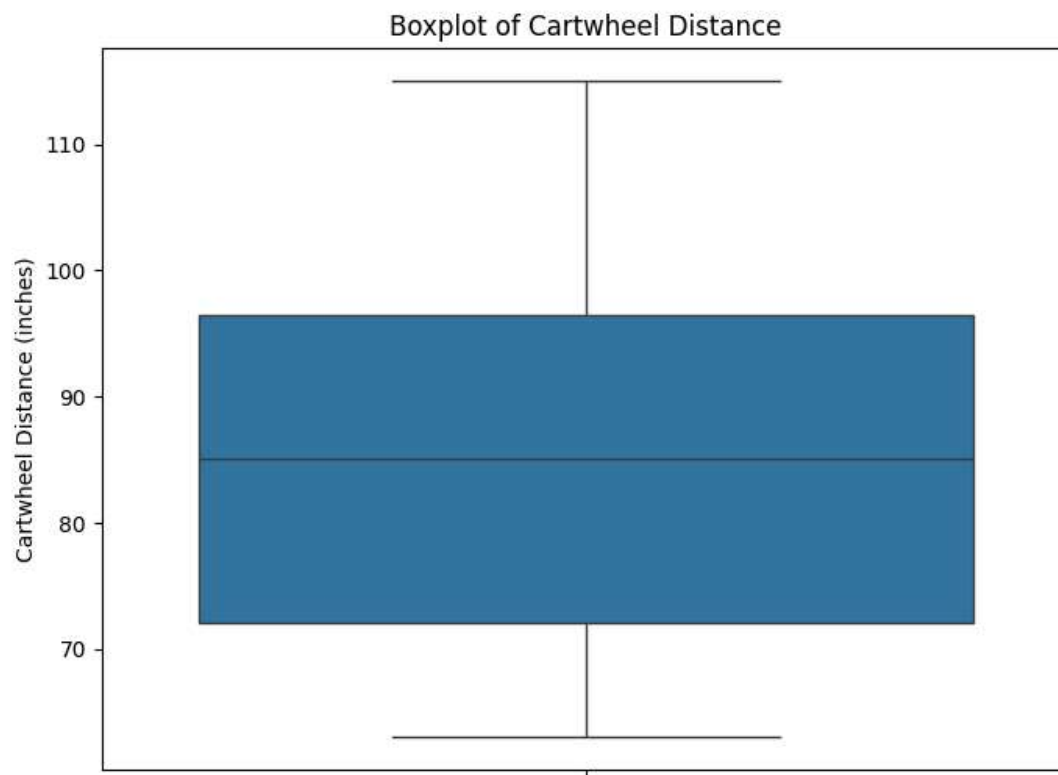
Boxplots do not show the shape of the distribution, but they can give us a better idea about the center and spread of the distribution as well as any potential outliers that may exist. Boxplots and Histograms often complement each other and help an analyst get more information about the data

```

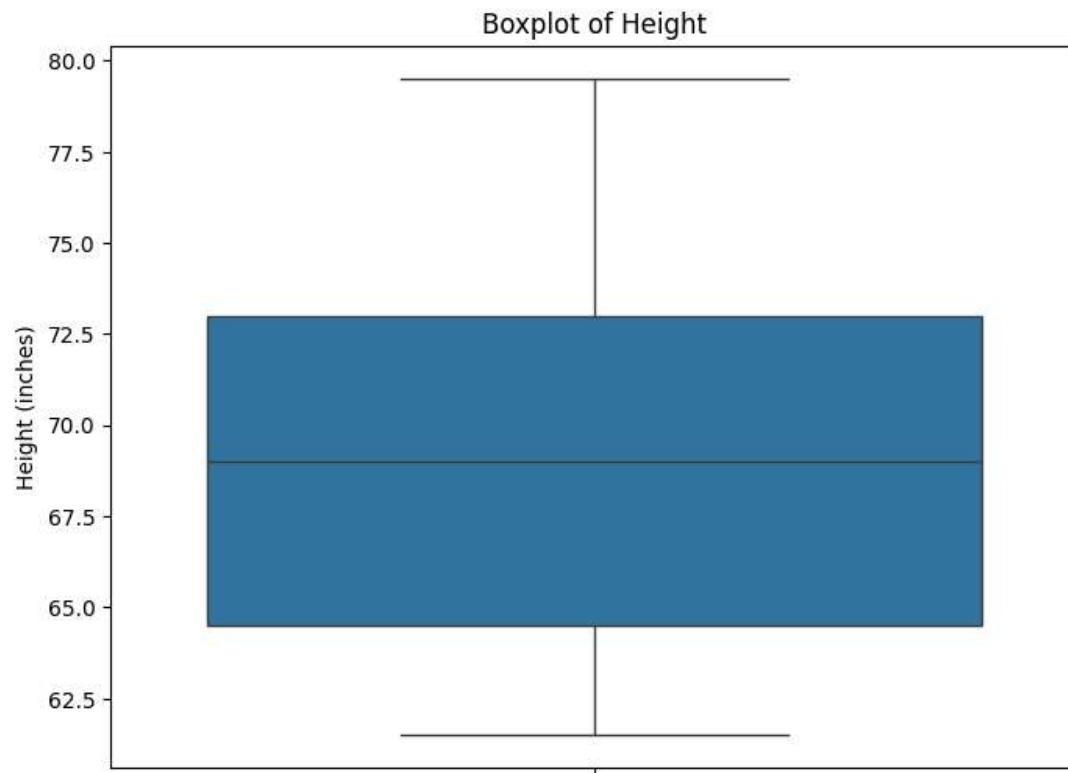
1 # Create the boxplot of the "CWDistance"
2 plt.figure(figsize=(8, 6))
3 sns.boxplot(data=df, y='CWDistance')
4
5 # Add titles and labels
6 plt.title('Boxplot of Cartwheel Distance')
7 plt.ylabel('Cartwheel Distance (inches)')
8
9 # Display the plot
10 plt.show()

```

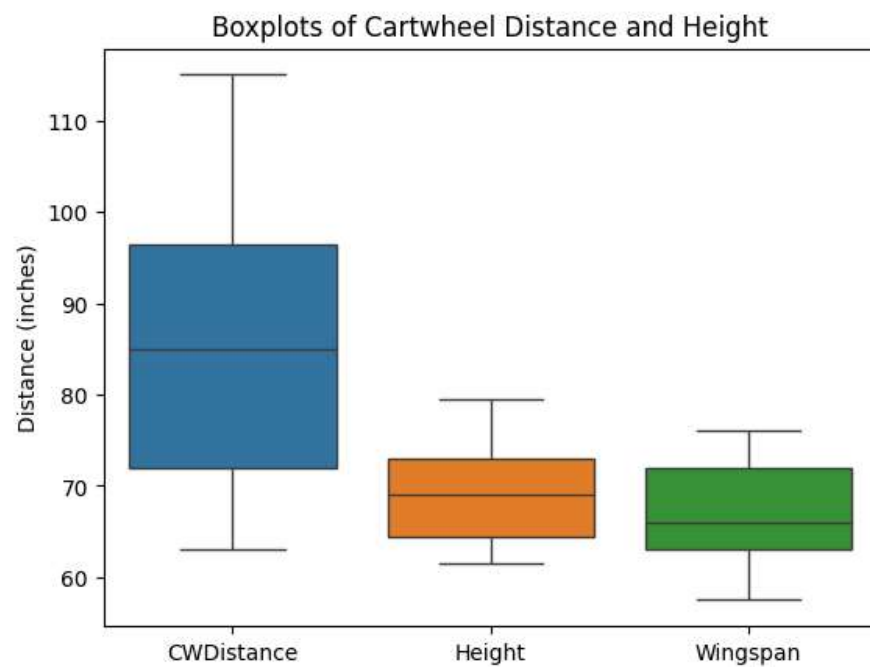




```
1 # Create the boxplot of the "Height"
2 plt.figure(figsize=(8, 6))
3 sns.boxplot(data=df, y='Height')
4
5 # Add titles and labels
6 plt.title('Boxplot of Height')
7 plt.ylabel('Height (inches)')
8
9 # Display the plot
10 plt.show()
```



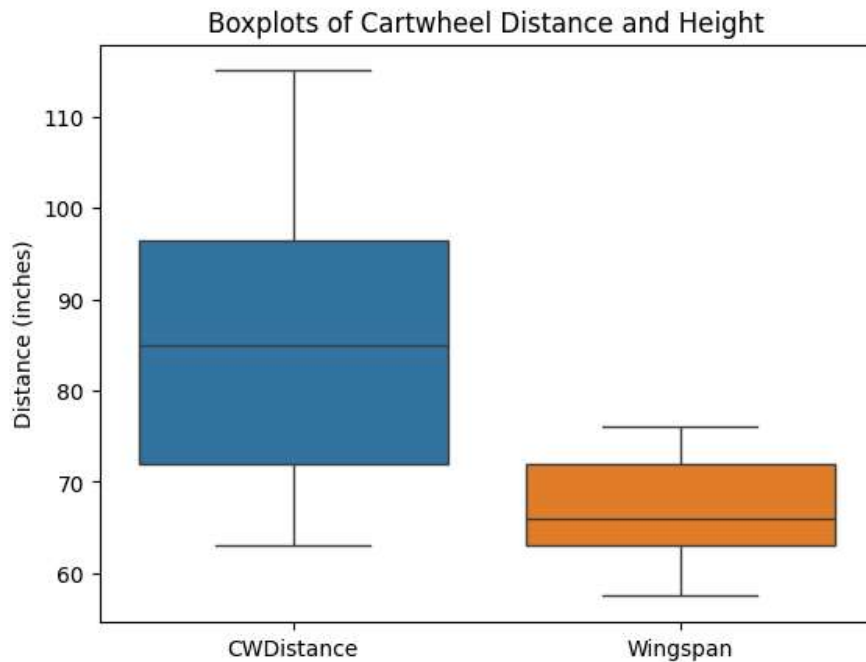
```
1 # Create the boxplots of the "CWDistance" and of the "Height" and of the "Wingspan"  
2 df_variables = df[['CWDistance', 'Height', 'Wingspan']]  
3 sns.boxplot(data=df_variables)  
4 plt.title('Boxplots of Cartwheel Distance and Height')  
5 plt.ylabel('Distance (inches)')  
6 plt.show()
```



```

1 # Create the boxplots of the "CWDistance" and of the "Wingspan"
2 df_variables = df[['CWDistance', 'Wingspan']]
3 sns.boxplot(data=df_variables)
4 plt.title('Boxplots of Cartwheel Distance and Height')
5 plt.ylabel('Distance (inches)')
6 plt.show()
7

```



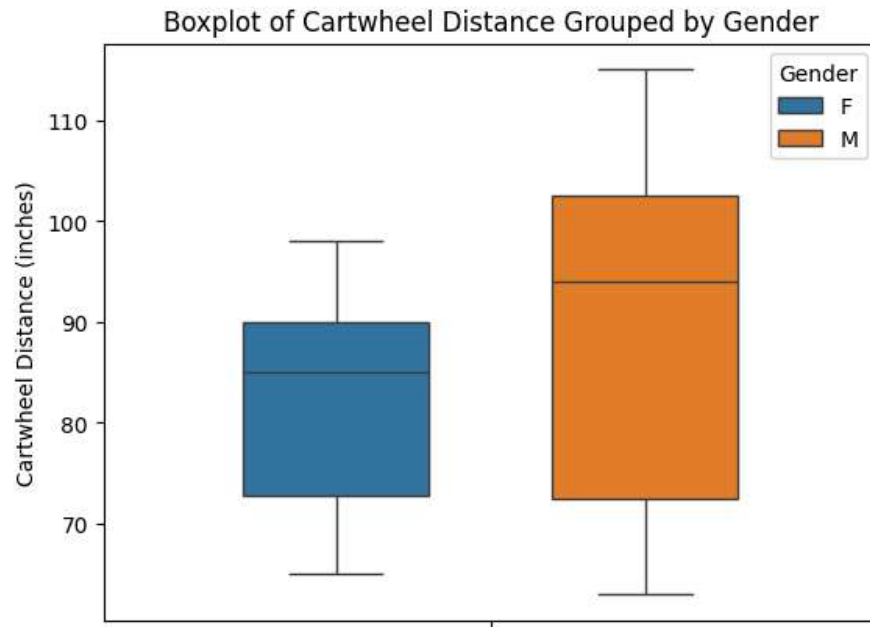
## ✓ Boxplots plotted by groups

While looking at a single variable is interesting, it is often useful to see how a variable changes in response to another. Thus, we can create a side-by-side boxplots of one quantitative variable grouped by another categorical variables.

```

1 # Create side-by-side boxplots of the "Height" grouped by "Gender"
2 # Create a boxplot of Height grouped by Gender
3 sns.boxplot(data=df, y='CWDistance', hue='Gender', gap=.4)
4 # Add titles and labels
5 plt.title('Boxplot of Cartwheel Distance Grouped by Gender')
6 plt.ylabel('Cartwheel Distance (inches)')
7 plt.legend(title='Gender') # Display the legend with the title 'Gender'
8 # Display the plot
9 plt.show()

```



## ✓ Histograms and boxplots plotted by groups

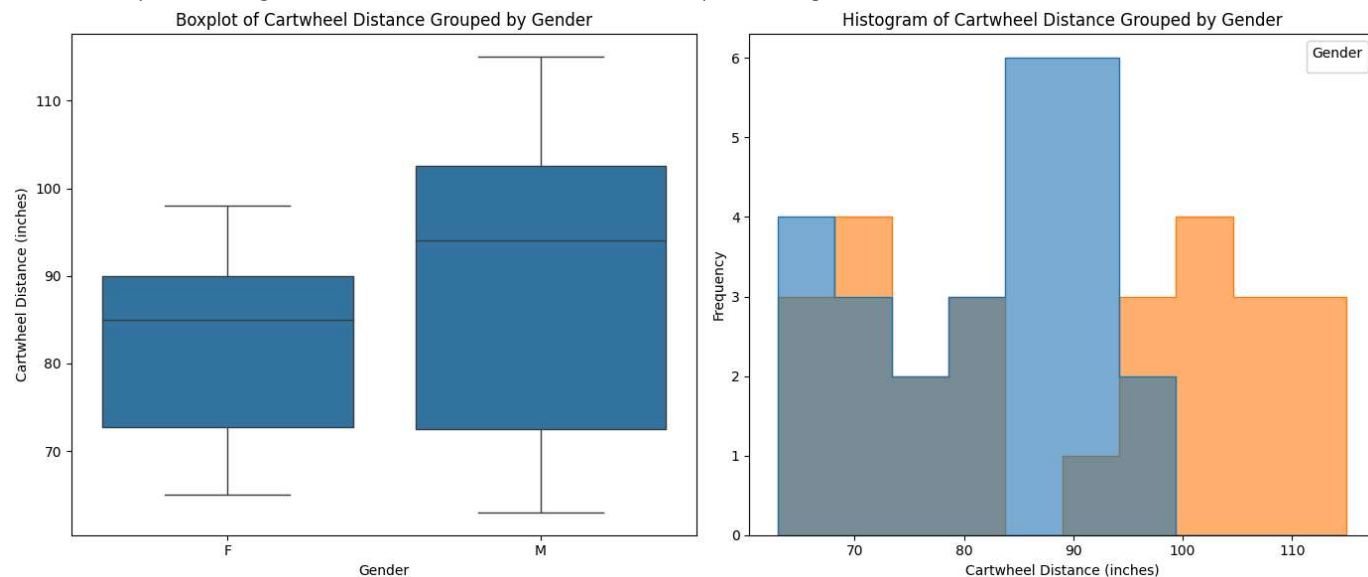
We can also create both boxplots and histograms of one quantitative variable grouped by another categorical variables

```

1 # Create a boxplot and histogram of the "CWDistance" grouped by "Gender"
2 plt.figure(figsize=(14, 6))
3
4 # Create the boxplot of CWDistance grouped by Gender
5 plt.subplot(1, 2, 1)
6 sns.boxplot(data=df, x='Gender', y='CWDistance')
7 plt.title('Boxplot of Cartwheel Distance Grouped by Gender')
8 plt.xlabel('Gender')
9 plt.ylabel('Cartwheel Distance (inches)')
10
11 # Create the histogram of CWDistance grouped by Gender
12 plt.subplot(1, 2, 2)
13 sns.histplot(data=df, x='CWDistance', hue='Gender', kde=False, bins=10, element='step', alpha=0.6)
14 plt.title('Histogram of Cartwheel Distance Grouped by Gender')
15 plt.xlabel('Cartwheel Distance (inches)')
16 plt.ylabel('Frequency')
17 plt.legend(title='Gender')
18
19 # Adjust the layout and display the plots
20 plt.tight_layout()
21 plt.show()

```

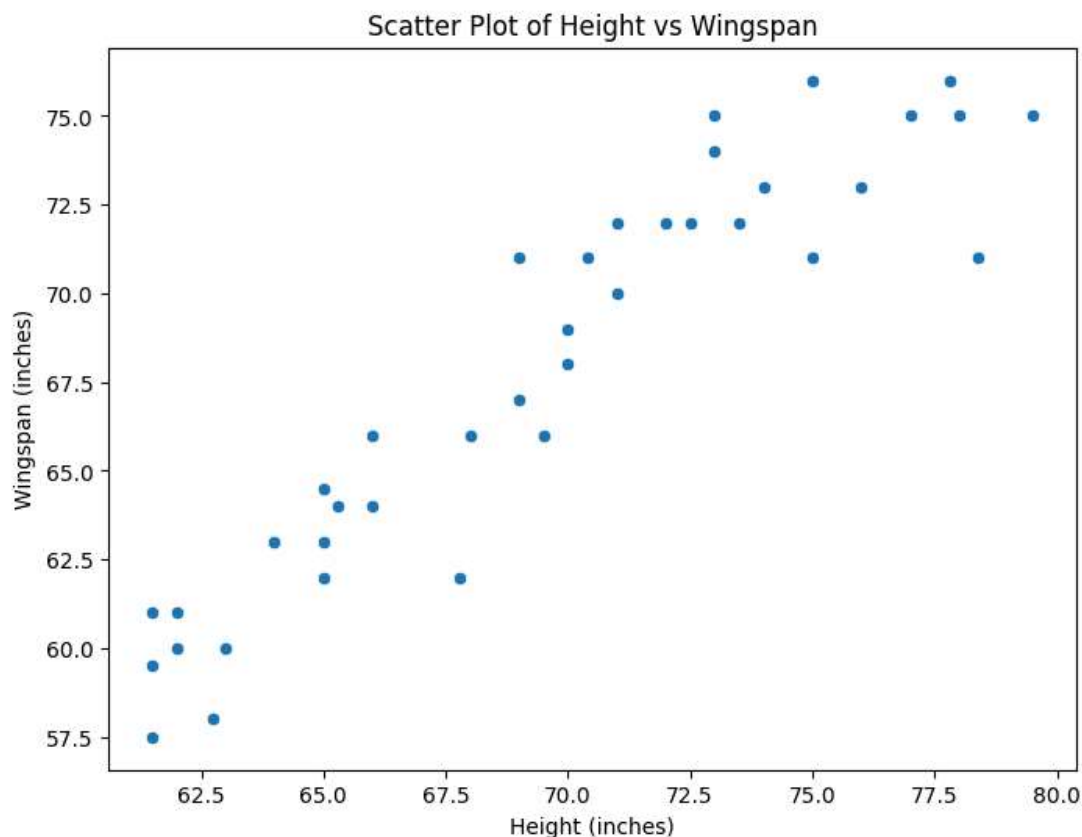
⚠ WARNING:matplotlib.legend.No artists with labels found to put in legend. Note that artists whose label start with



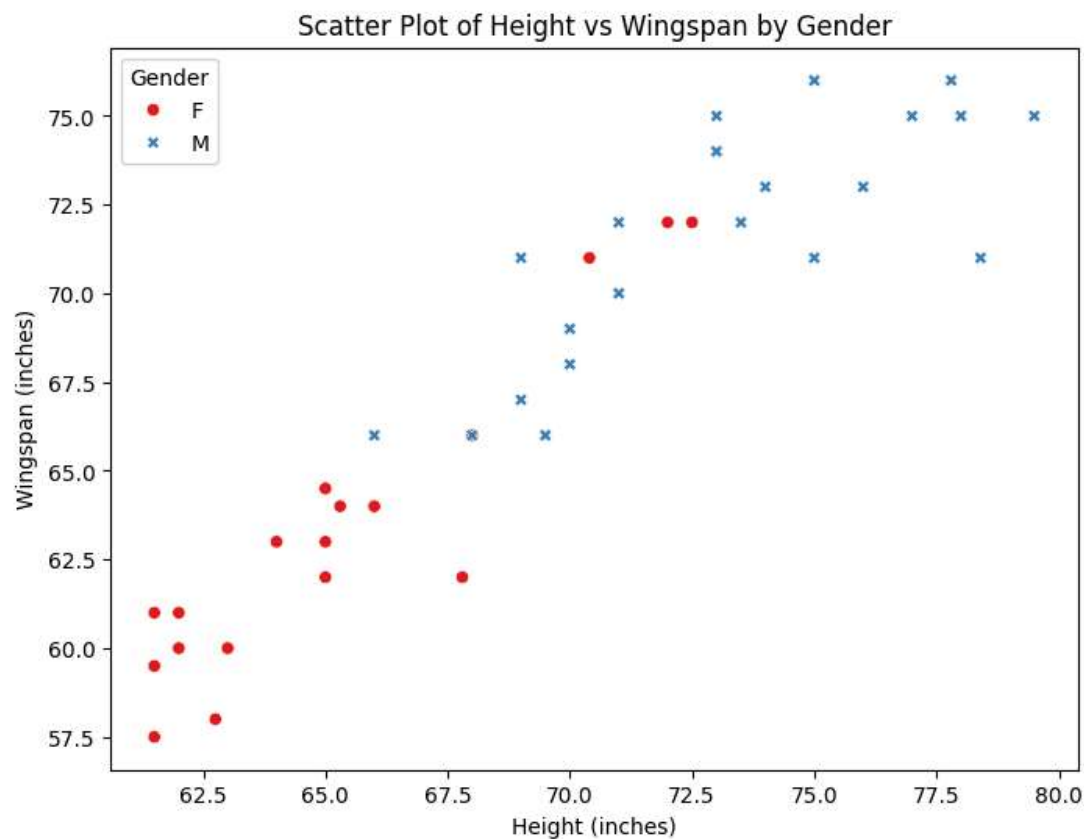
## ✓ Scatter plot

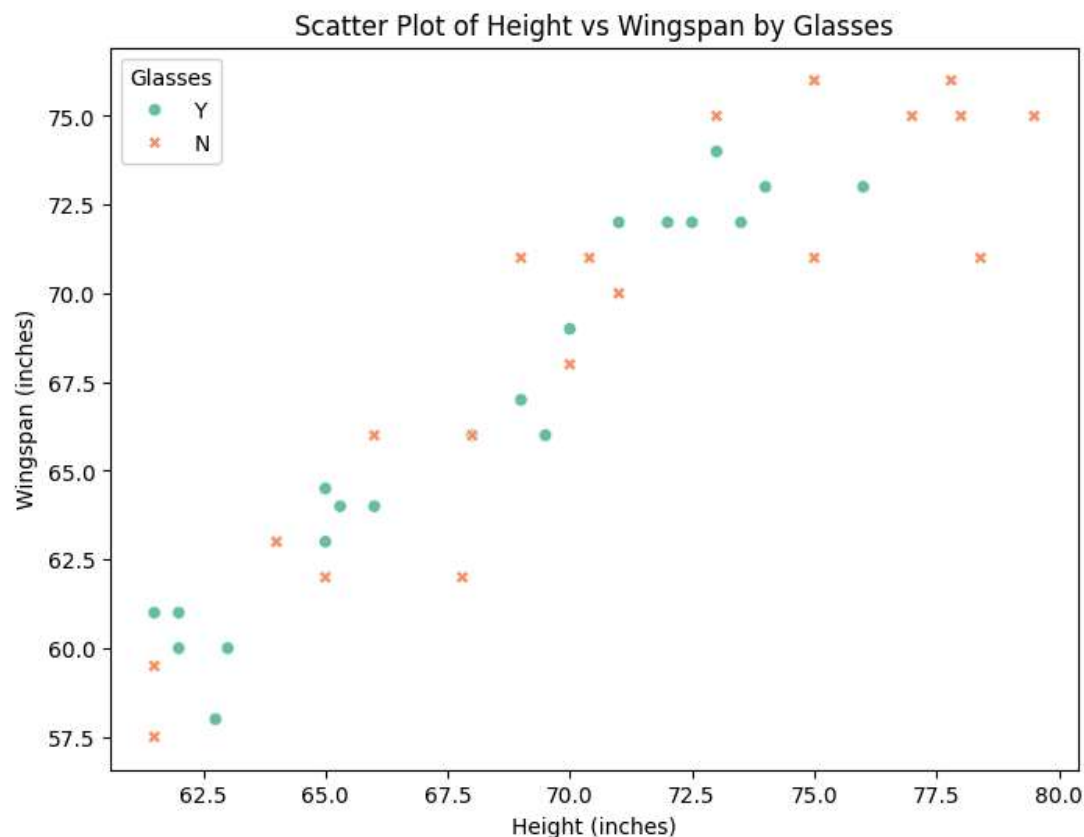
Plot values of one variable versus another variable to see how they are correlated

```
1 # scatter plot between two variables
2 # Scatter plot of Height vs Wingspan
3 plt.figure(figsize=(8, 6))
4 sns.scatterplot(data=df, x='Height', y='Wingspan')
5 plt.title('Scatter Plot of Height vs Wingspan')
6 plt.xlabel('Height (inches)')
7 plt.ylabel('Wingspan (inches)')
8 plt.show()
9
```



```
1 # scatter plot between two variables (one categorical)
2 # Scatter plot of Height vs Wingspan with Gender as hue
3 plt.figure(figsize=(8, 6))
4 sns.scatterplot(data=df, x='Height', y='Wingspan', hue='Gender', style='Gender', palette='Set1')
5 plt.title('Scatter Plot of Height vs Wingspan by Gender')
6 plt.xlabel('Height (inches)')
7 plt.ylabel('Wingspan (inches)')
8 plt.legend(title='Gender')
9 plt.show()
```



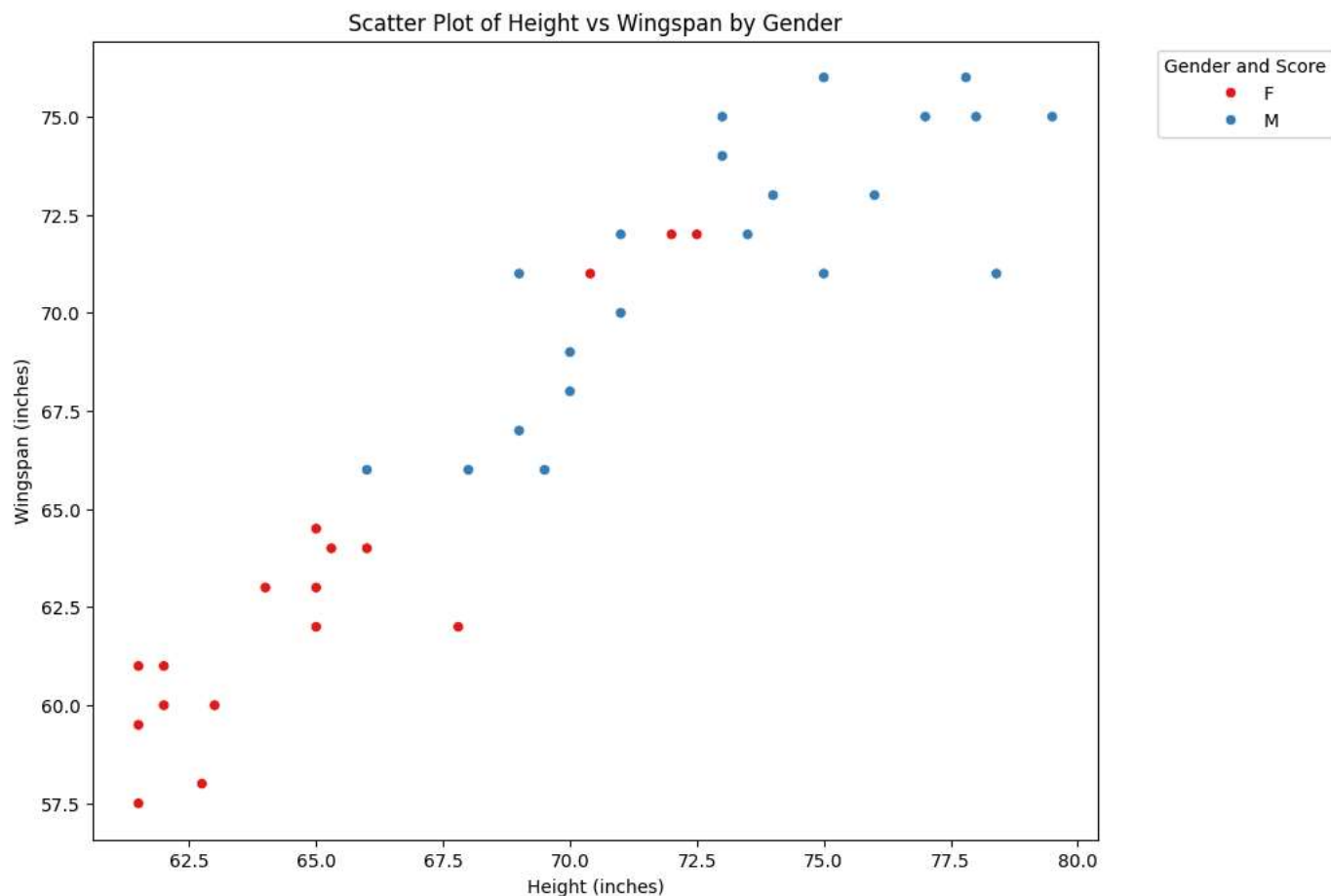


```

1 # Scatter plot between two variables grouped according to a categorical variable
2 plt.figure(figsize=(10, 8))
3 sns.scatterplot(data=df, x='Height', y='Wingspan', hue='Gender', palette='Set1', legend='brief')
4 plt.title('Scatter Plot of Height vs Wingspan by Gender')
5 plt.xlabel('Height (inches)')
6 plt.ylabel('Wingspan (inches)')
7 plt.legend(title='Gender and Score', bbox_to_anchor=(1.05, 1), loc='upper left')
8 plt.show()

```





```

1 # scatter plot between two variables grouped according to a categorical variable and with size of markers
2 # Scatter plot of Height vs Wingspan with Gender as hue and Size of markers by Score
3 plt.figure(figsize=(10, 8))
4 sns.scatterplot(data=df, x='Height', y='Wingspan', hue='Gender', size='Score', palette='Set1', sizes=(20, 200), lege
5 plt.title('Scatter Plot of Height vs Wingspan by Gender with Score as Marker Size')
6 plt.xlabel('Height (inches)')
7 plt.ylabel('Wingspan (inches)')
8 plt.legend(title='Gender and Score', bbox_to_anchor=(1.05, 1), loc='upper left')
9 plt.show()
10

```