Cheatsheet - Seaborn Charts

This kernel is all about seaborn charts -- python data visualization tool.

Mohammad Ashour

- linkedin : https://www.linkedin.com/in/mohammad-ashour-33145a204/
- youtube : https://www.youtube.com/channel/UC8qRDUu1sSk5gGA_fJug2ZA
- medium : https://medium.com/@mohamad.ashour203

Table of Content

Introduction

Seaborn

Seaborn is a graphic library built on top of Matplotlib. It allows to make your charts prettier, and facilitates some of the common data visualisation needs. What so special about seaborn? Why do we need to use seaborn while we already have Maplotlib? Matplotlib can serve your purpose. It has all the visualization that you need to perform a data storytelling project. But seaborn is special because it comes in with a lot of styles. The style is already built-in. Compared to an ordinary matplotlib plot, an ordinary seaborn plot look a lot nicer!

Libraries

In [1]:
```
# Import dependencies

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
```

Load Data

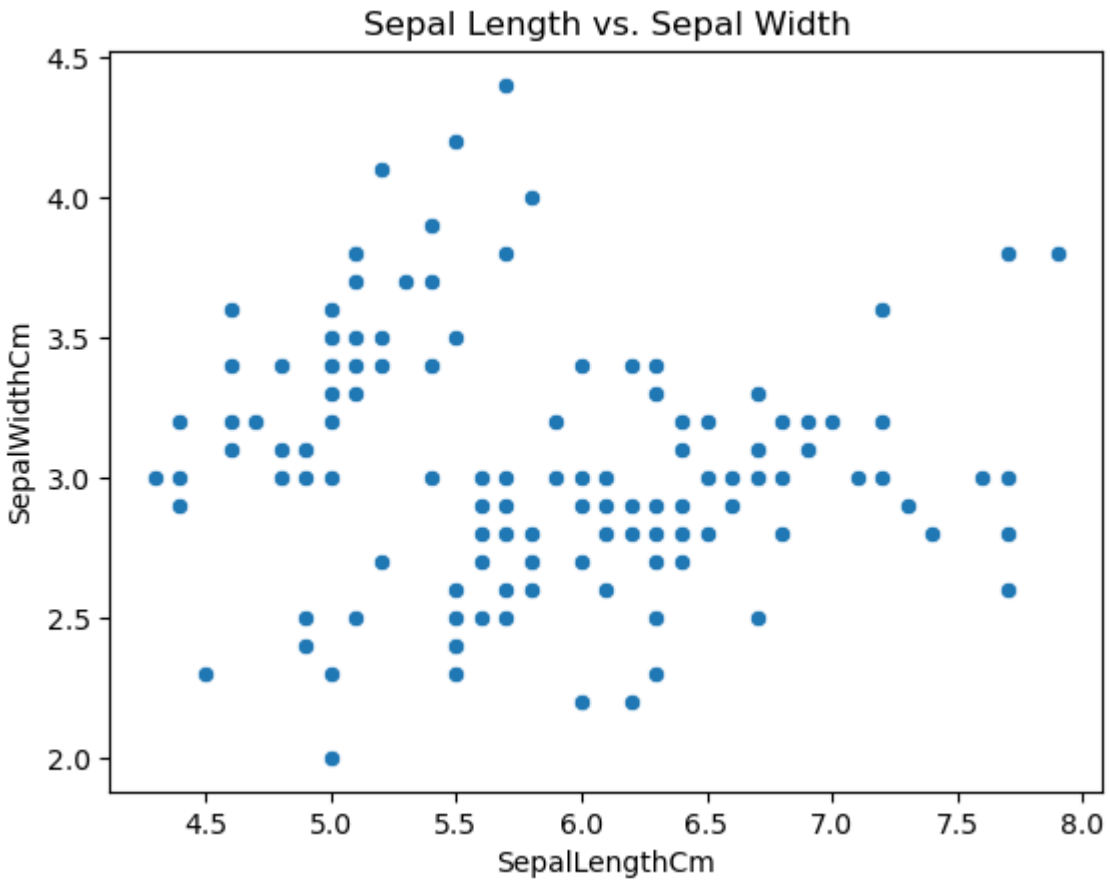
In [2]:
```
olympic = pd.read_csv('../input/120-years-of-olympic-history-athletes-and-results/at
healthcare=pd.read_csv('../input/av-healthcare-analytics-ii/healthcare/train_data.cs
covid_india=pd.read_csv('../input/covid19-in-india/covid_19_india.csv')
google_playstore=pd.read_csv('../input/google-play-store-apps/googleplaystore.csv')
iris=pd.read_csv('../input/iris/Iris.csv')
corona_virus = pd.read_csv('../input/novel-corona-virus-2019-dataset/covid_19_data.c
youtube=pd.read_csv('../input/youtube-new/USvideos.csv')
shootings=pd.read_csv('../input/us-police-shootings/shootings.csv')
```

Scatter Plot

Basic Scatter Plot

In [3]:
```
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm', data=iris)
plt.title('Sepal Length vs. Sepal Width')
```

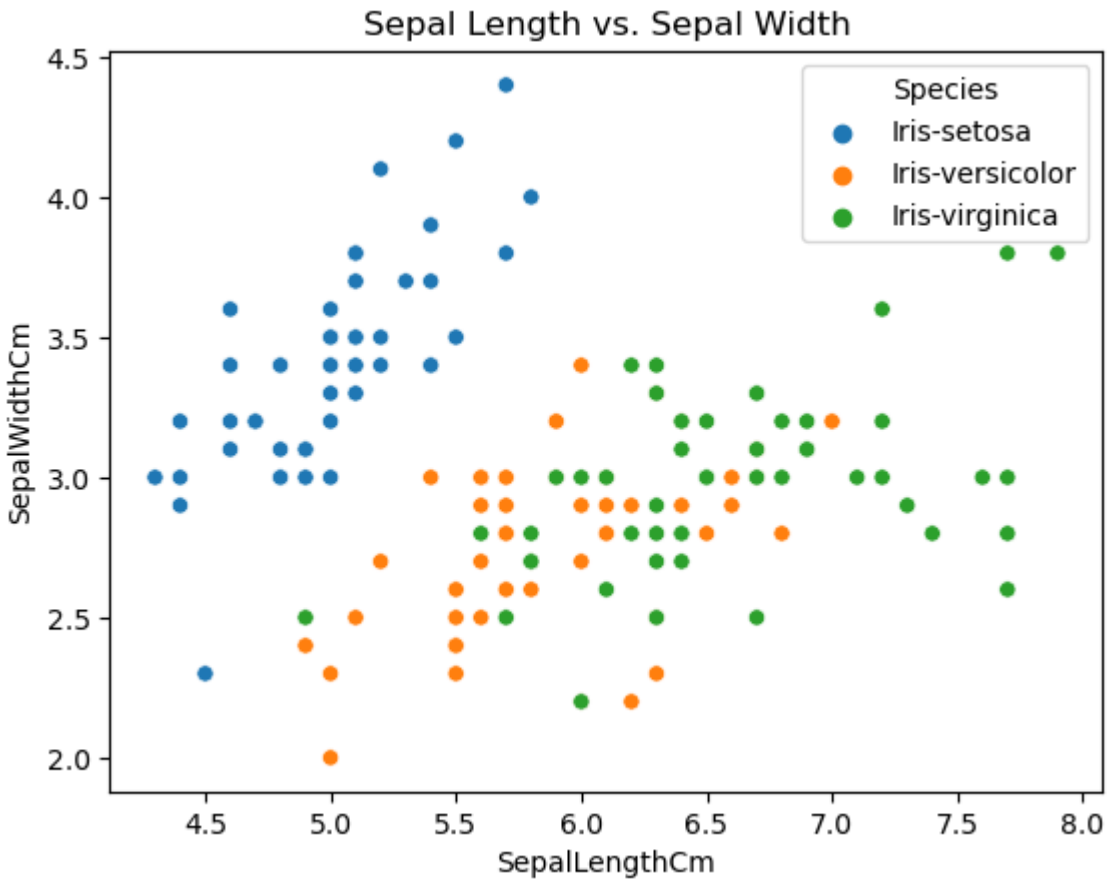
Out[3]:
```
Text(0.5, 1.0, 'Sepal Length vs. Sepal Width')
```



Scatter Plot - Category

In [4]:
```
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm', hue='Species', data=iris)
plt.title('Sepal Length vs. Sepal Width')
```

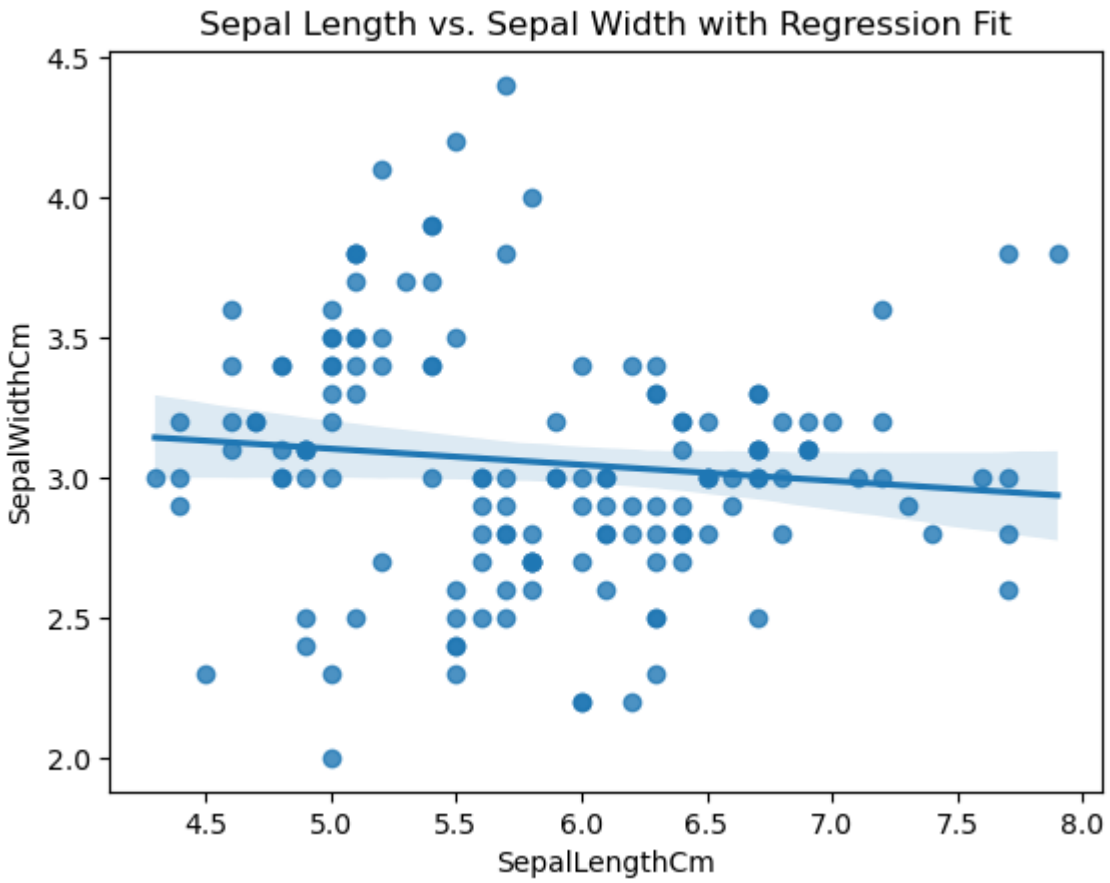
Out[4]:
```
Text(0.5, 1.0, 'Sepal Length vs. Sepal Width')
```



Scatter Plot with Regression Fit

In [5]:
```
sns.regplot(x='SepalLengthCm', y='SepalWidthCm', data=iris)
plt.title('Sepal Length vs. Sepal Width with Regression Fit')
```

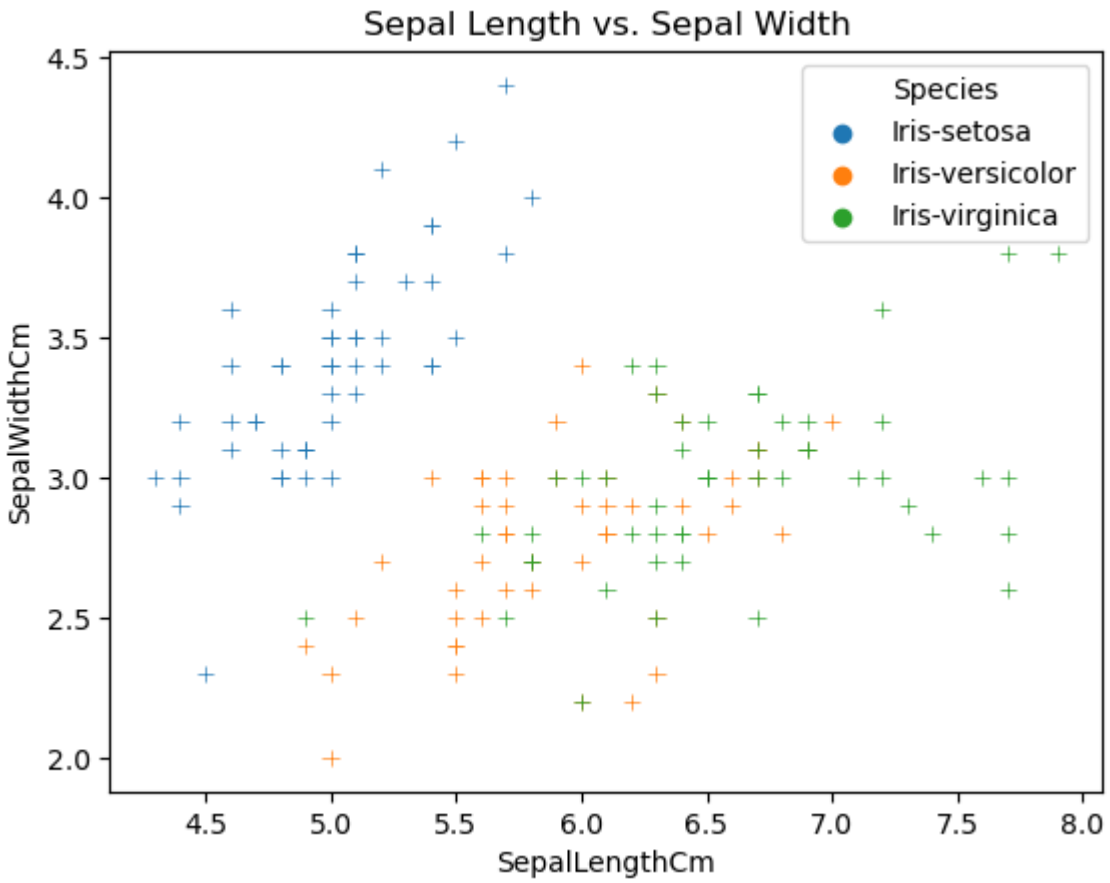
Out[5]:
```
Text(0.5, 1.0, 'Sepal Length vs. Sepal Width with Regression Fit')
```



Control shape of element in Scatter Plot

In [6]:
```
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm', hue='Species', data=iris, marke
plt.title('Sepal Length vs. Sepal Width')
```

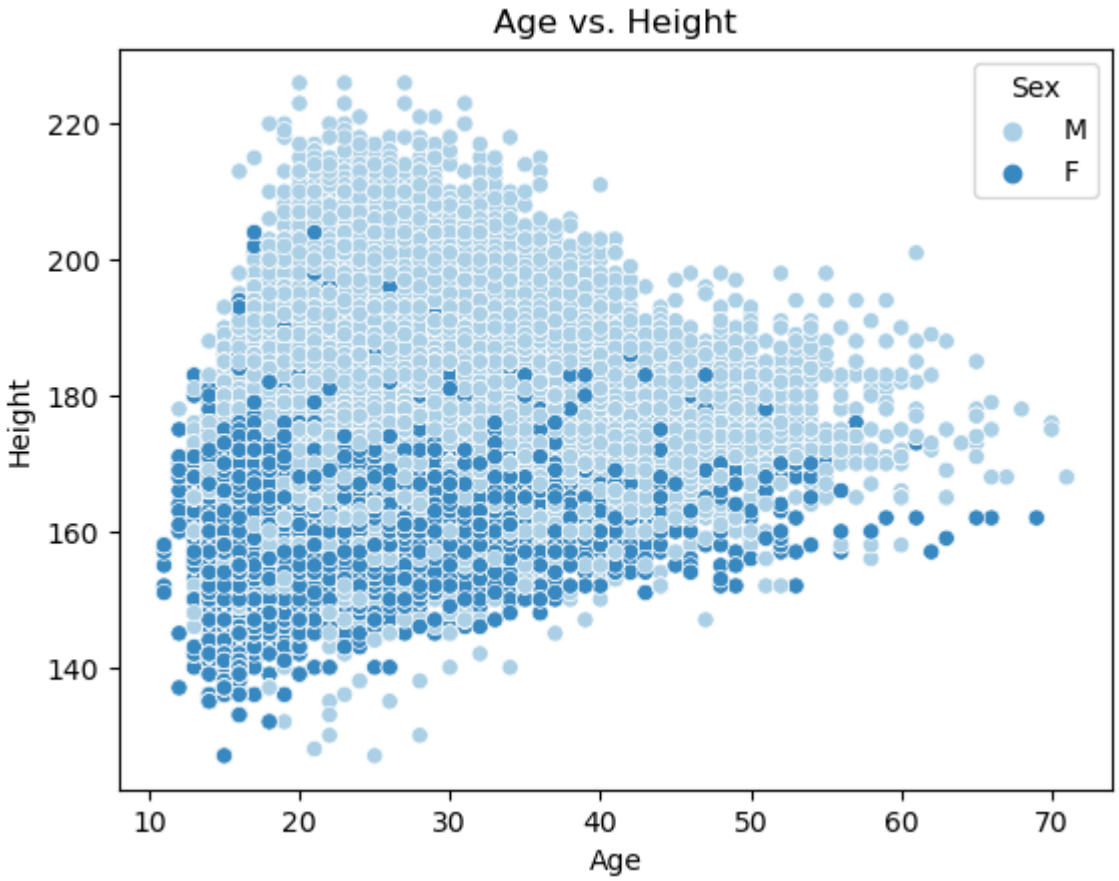
Out[6]:
```
Text(0.5, 1.0, 'Sepal Length vs. Sepal Width')
```



Using color Palletes in Scatter Plot

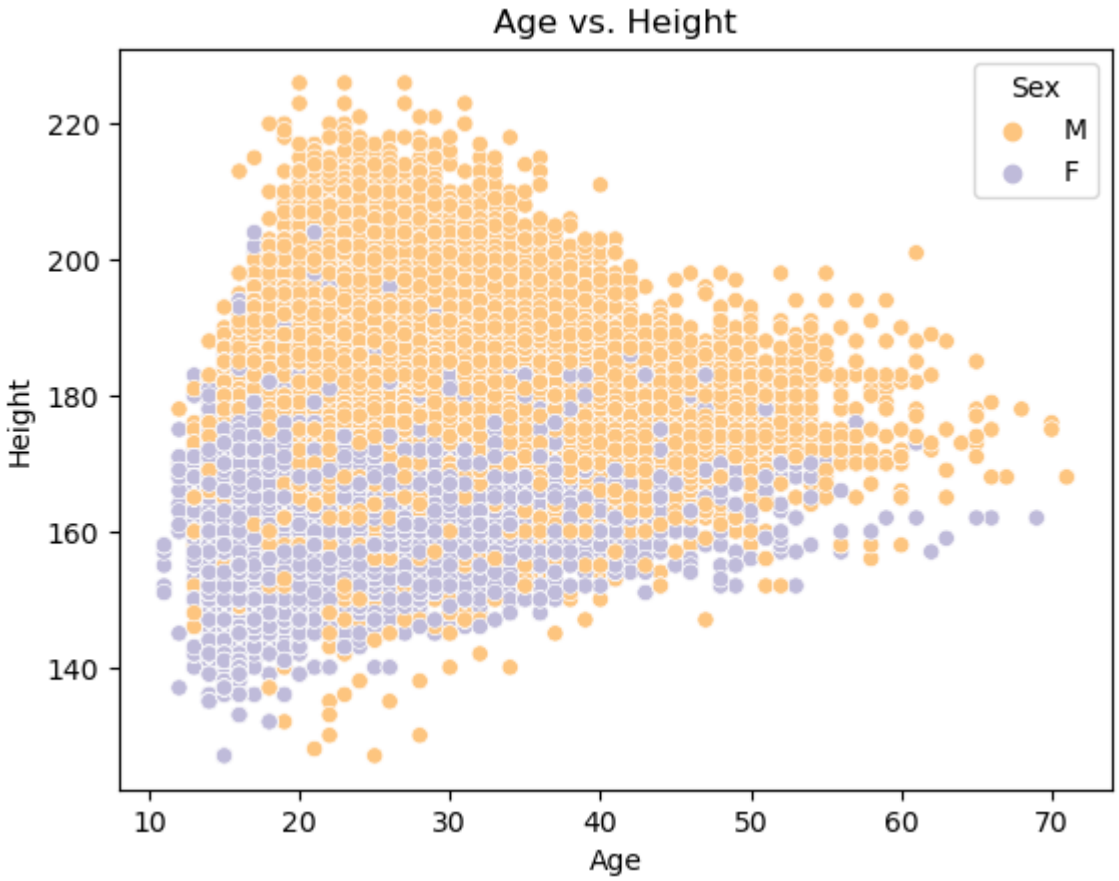
In [7]:
```
# using sequencial color scheme
sns.scatterplot(x='Age', y='Height', hue='Sex', data=olympic, palette='Blues')
plt.title('Age vs. Height')
```

Out[7]:
```
Text(0.5, 1.0, 'Age vs. Height')
```



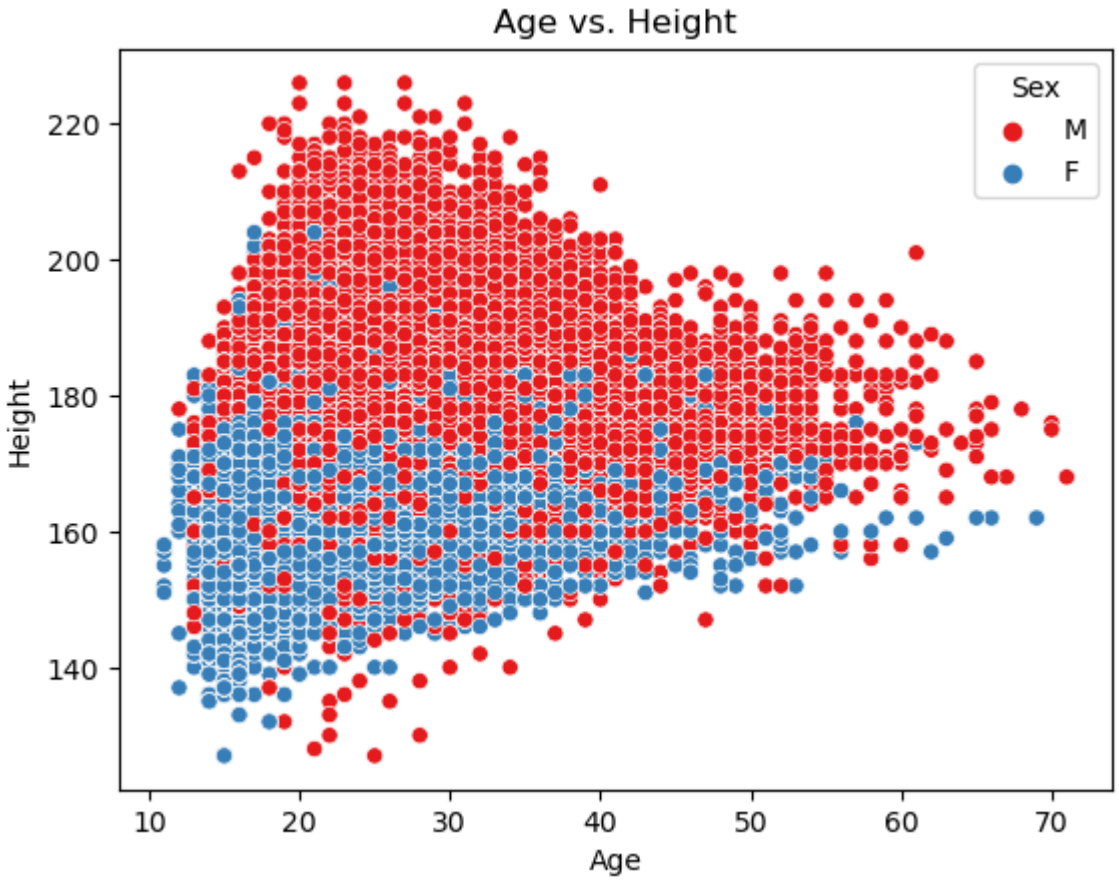
In [8]:
```
# using diverging color scheme
sns.scatterplot(x='Age', y='Height', hue='Sex', data=olympic, palette='PuOr')
plt.title('Age vs. Height')
```

Out[8]:
```
Text(0.5, 1.0, 'Age vs. Height')
```



In [9]:
```
# using discrete color scheme
sns.scatterplot(x='Age', y='Height', hue='Sex', data=olympic, palette='Set1')
plt.title('Age vs. Height')
```

Out[9]:
```
Text(0.5, 1.0, 'Age vs. Height')
```

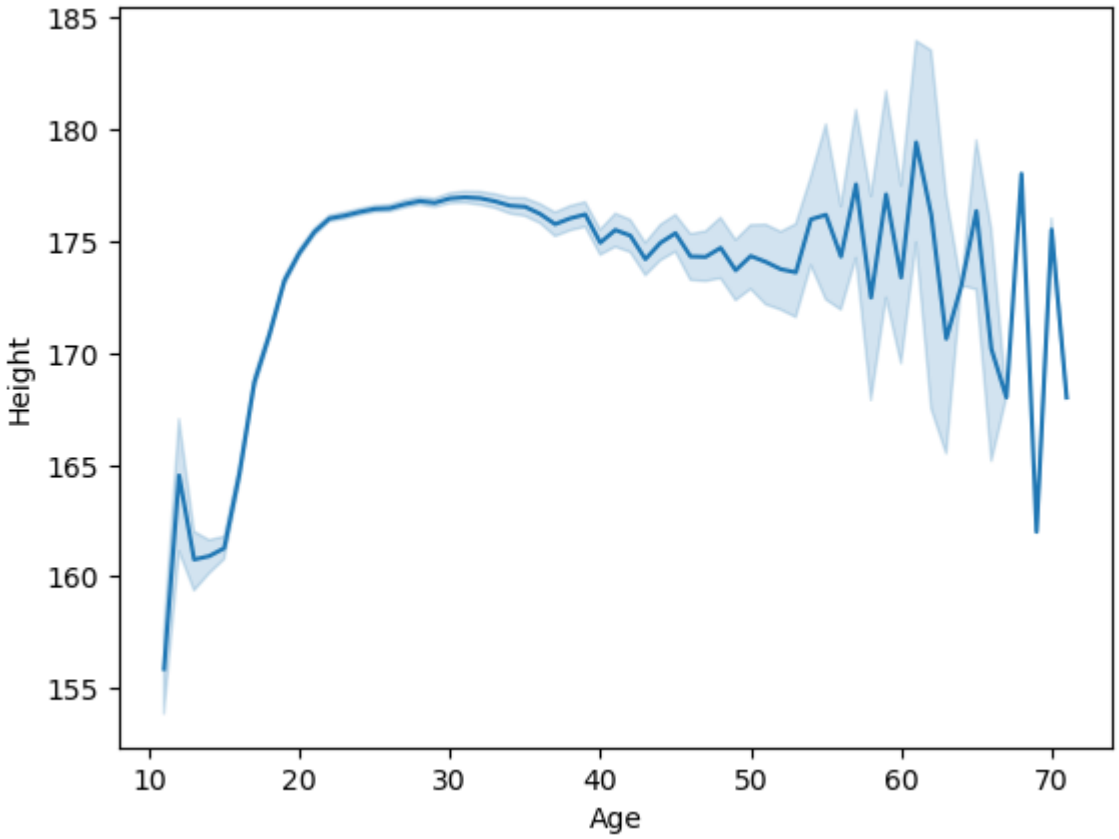


Line Plot

Basic Line Plot

In [10]:
```
sns.lineplot(x='Age', y='Height', data=olympic)
```

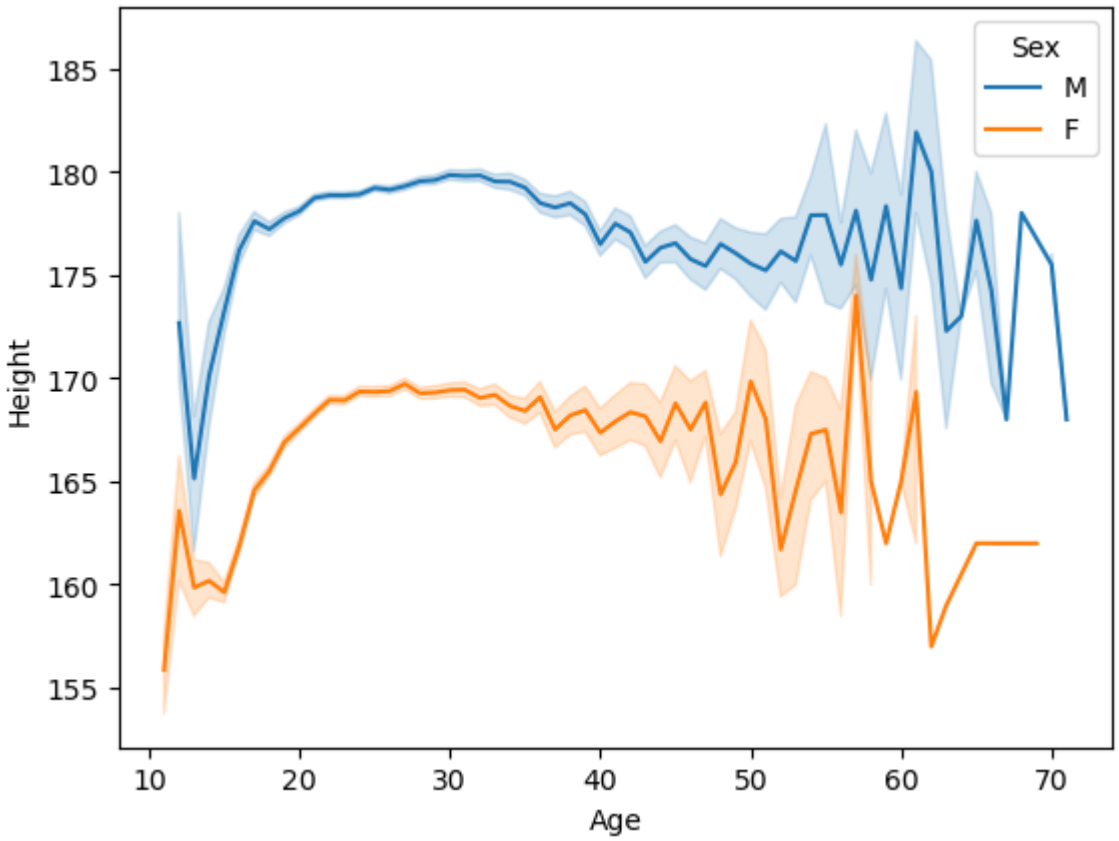
Out[10]:
```
<AxesSubplot:xlabel='Age', ylabel='Height'>
```



Line Plot - Category

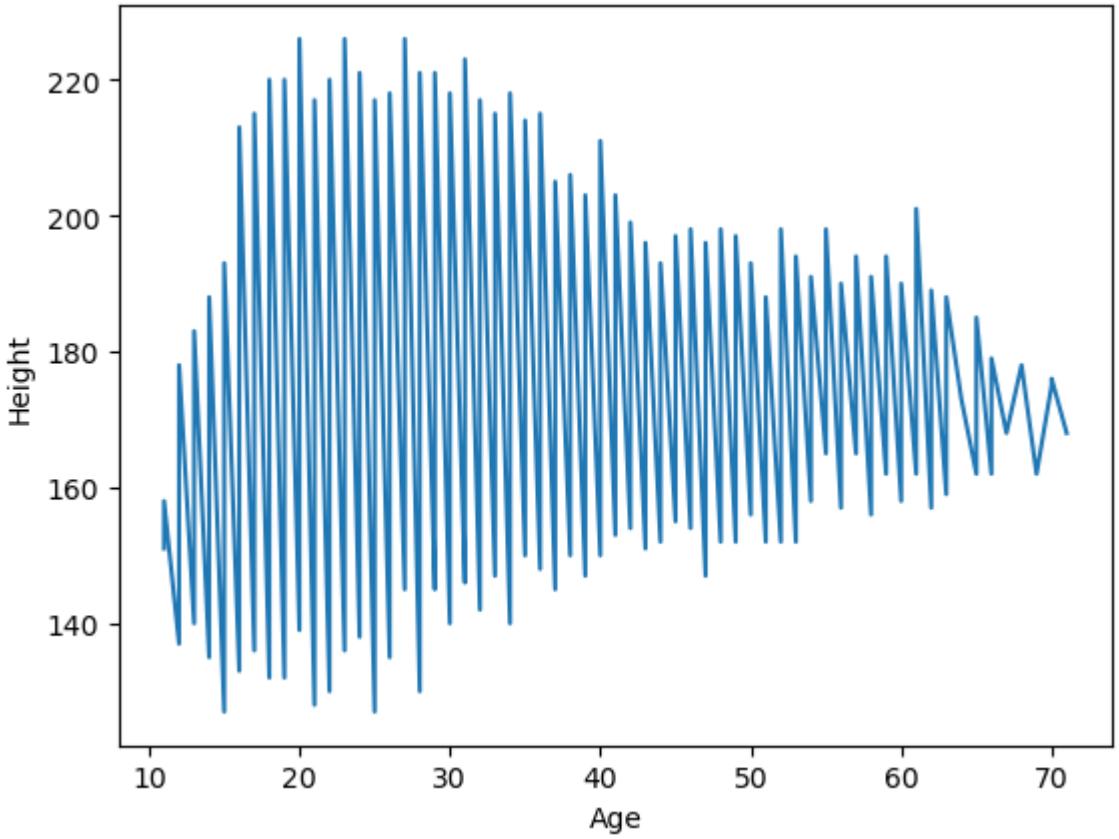
In [11]:
```
sns.lineplot(x='Age', y='Height', data=olympic, hue='Sex')
```

Out[11]:
```
<AxesSubplot:xlabel='Age', ylabel='Height'>
```



In [12]:
```
sns.lineplot(x='Age', y='Height', data=olympic, estimator=None)
```

Out[12]:
```
<AxesSubplot:xlabel='Age', ylabel='Height'>
```



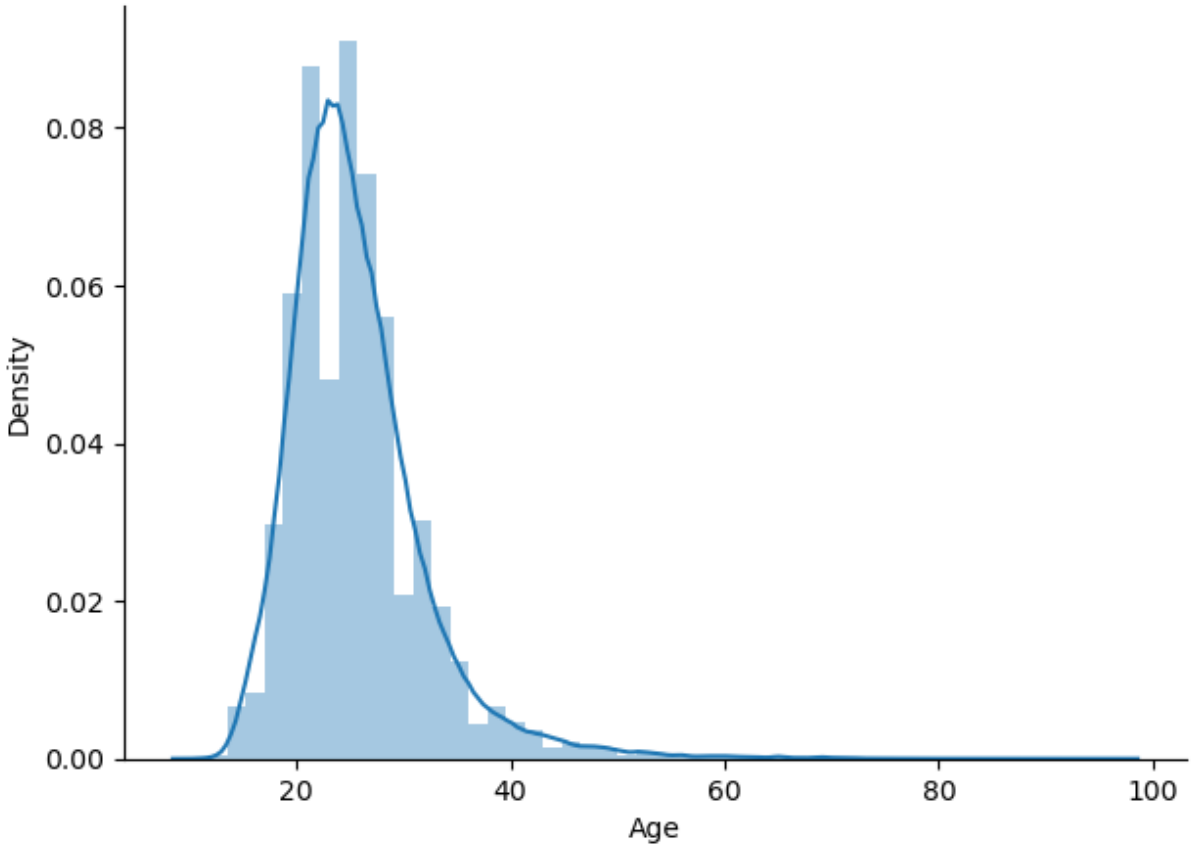
Histogram Plot

Basic Histogram Plot

In [13]:
```
f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)
sns.distplot(olympic['Age'])
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[13]:
```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```

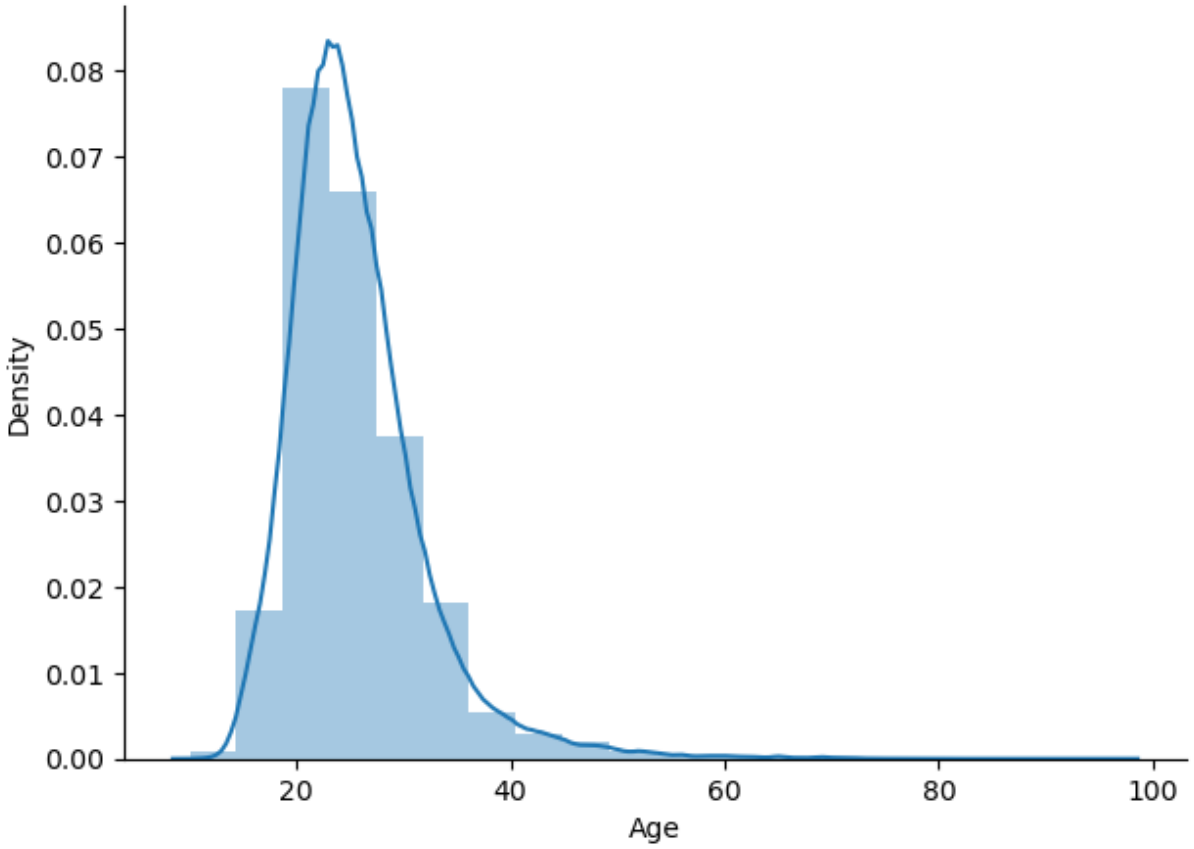


Histogram Plot : Control Number of Bins

In [14]:
```
f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)
sns.distplot(olympic['Age'], bins=20)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[14]:
```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```

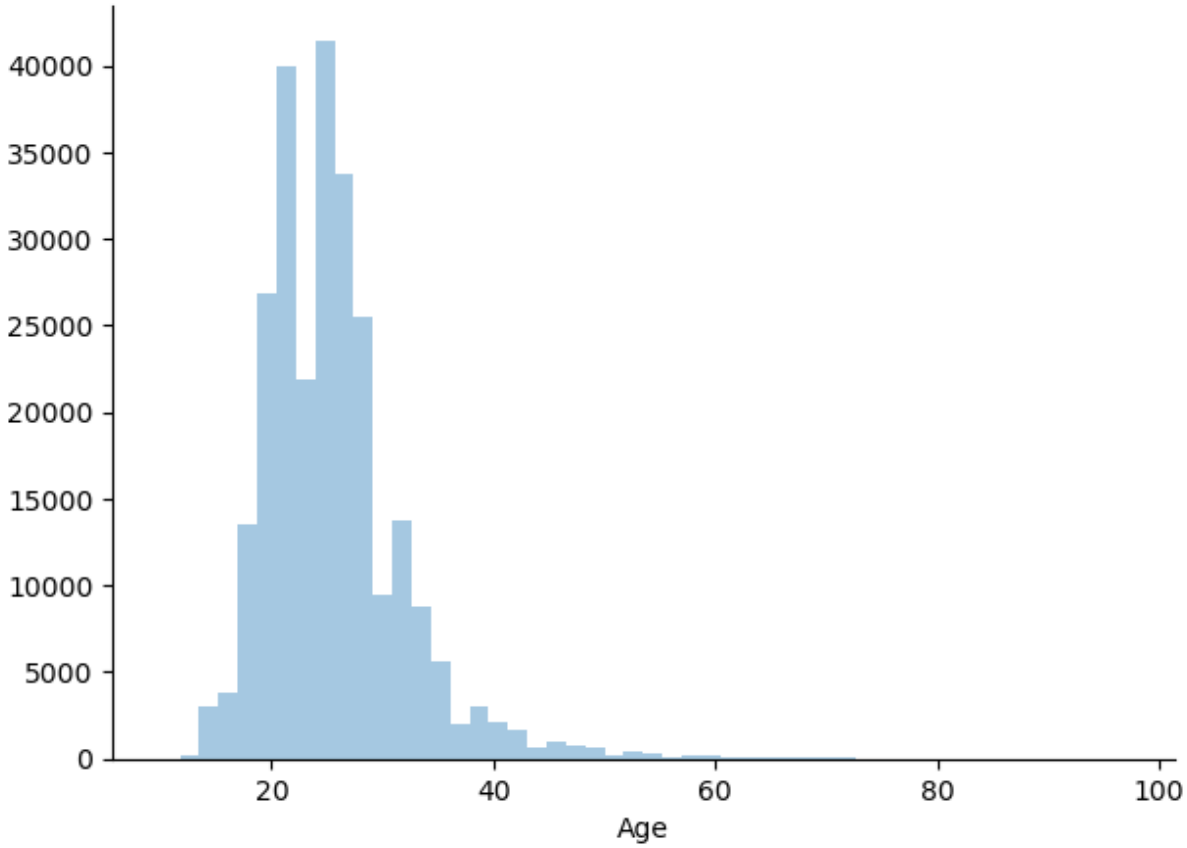


Histogram Plot : Control Density

In [15]:
```
f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)
sns.distplot(olympic['Age'], kde=False)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[15]: <AxesSubplot:xlabel='Age'>

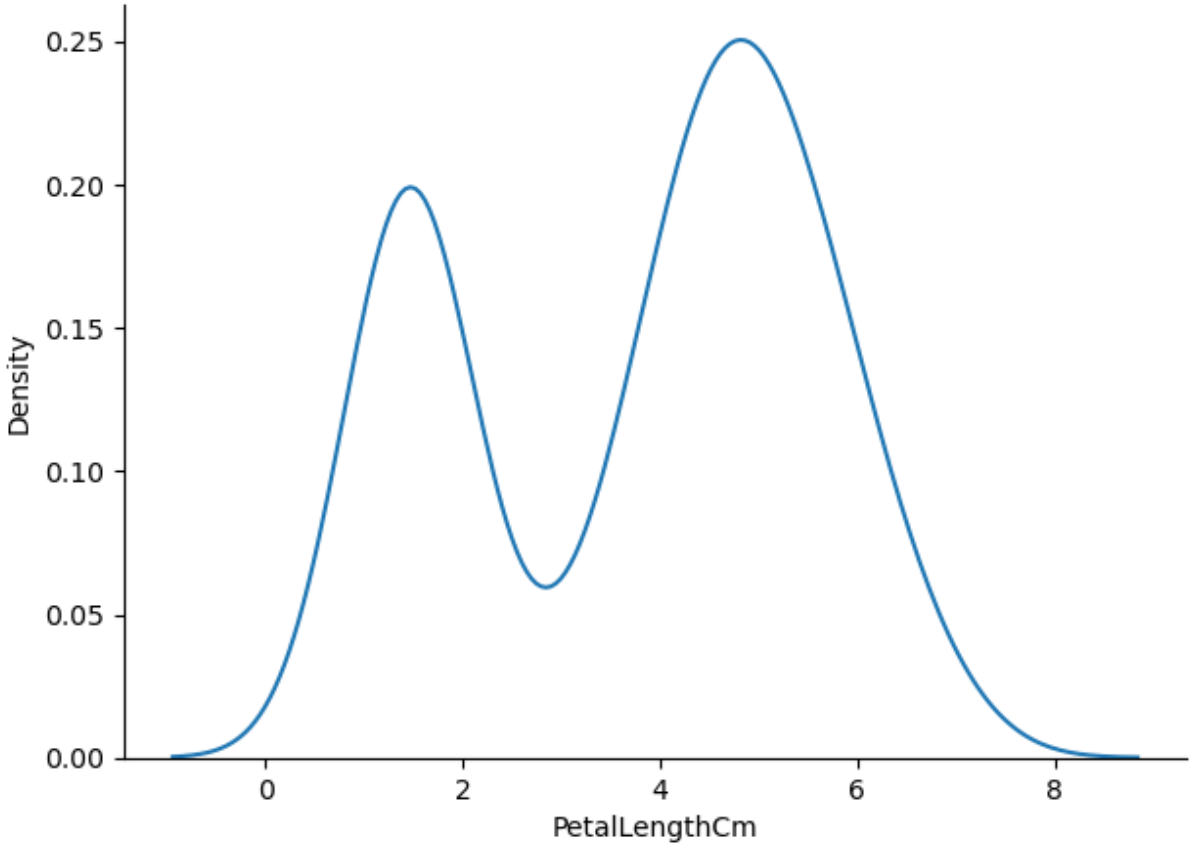


1D Density Plot

Basic 1D Density Plot

In [16]:
```
f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)
sns.kdeplot(iris['PetalLengthCm'])
```

Out[16]: <AxesSubplot:xlabel='PetalLengthCm', ylabel='Density'>

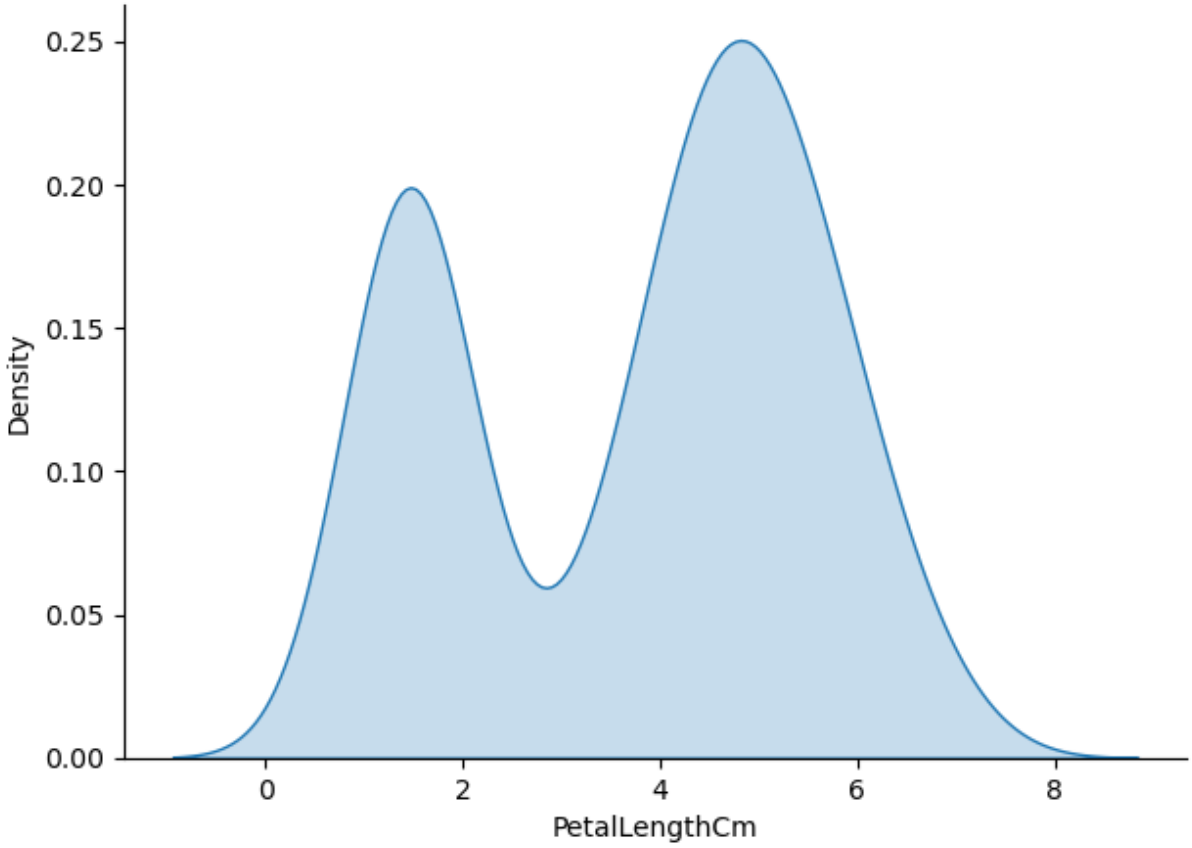


Density Plot with Shade

In [17]:
```
f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)
sns.kdeplot(iris['PetalLengthCm'], shade=True)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[17]:
```
<AxesSubplot:xlabel='PetalLengthCm', ylabel='Density'>
```

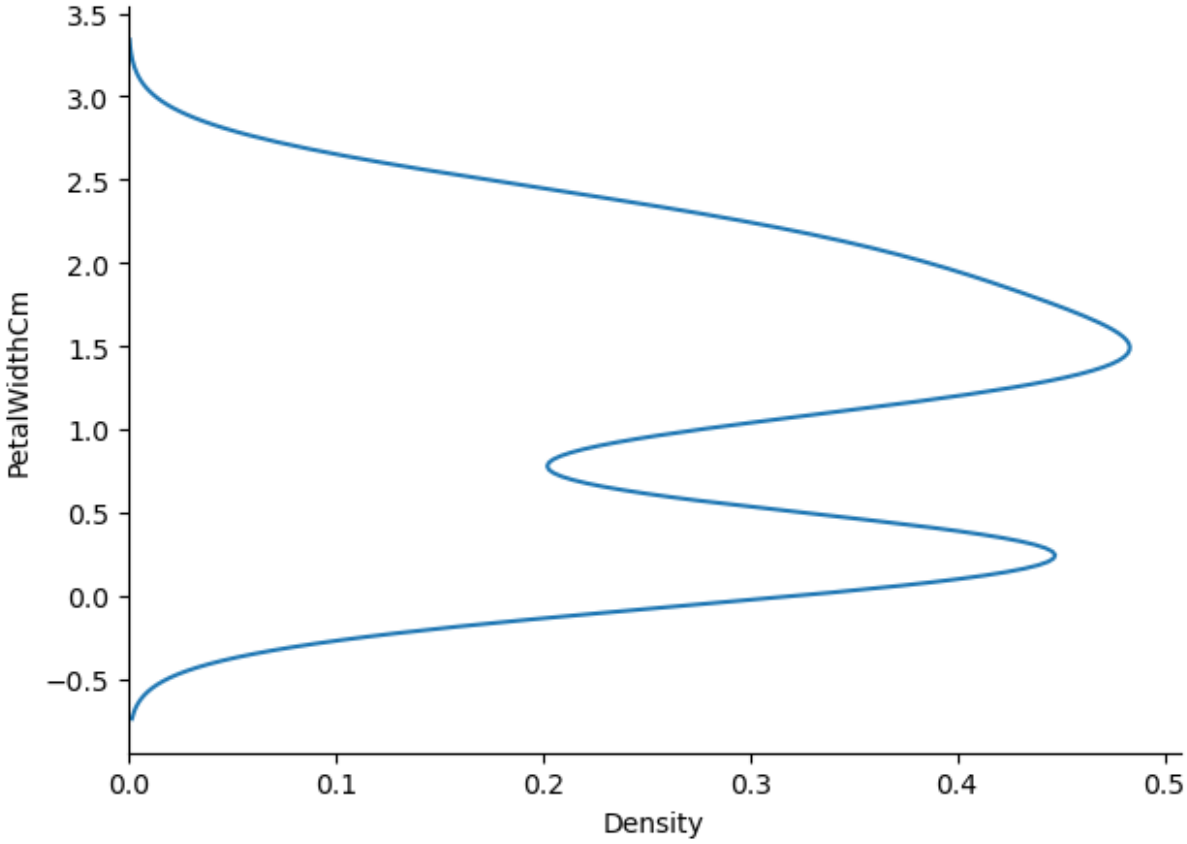


Horizontal Density Plot

In [18]:
```
f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)
sns.kdeplot(iris['PetalWidthCm'], vertical=True)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: UserWarning:

The `vertical` parameter is deprecated; assigning data to `y`.
This will become an error in seaborn v0.13.0; please update your code.

  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[18]:
```
<AxesSubplot:xlabel='Density', ylabel='PetalWidthCm'>
```

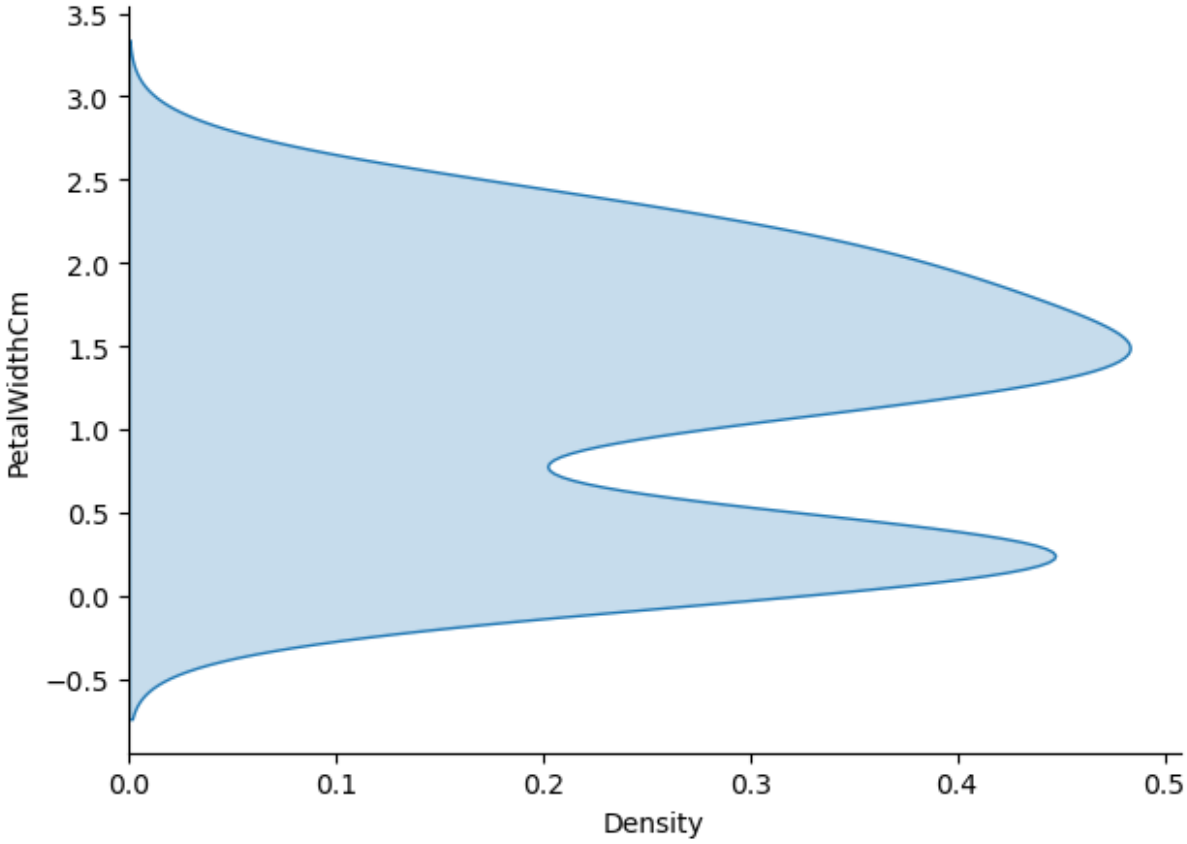


Horizontal Density Plot with Shade

In [19]:
```
f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)
sns.kdeplot(iris['PetalWidthCm'], vertical=True, shade=True)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: UserWarning:

The `vertical` parameter is deprecated; assigning data to `y`.
This will become an error in seaborn v0.13.0; please update your code.

  This is separate from the ipykernel package so we can avoid doing imports until
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  This is separate from the ipykernel package so we can avoid doing imports until
```

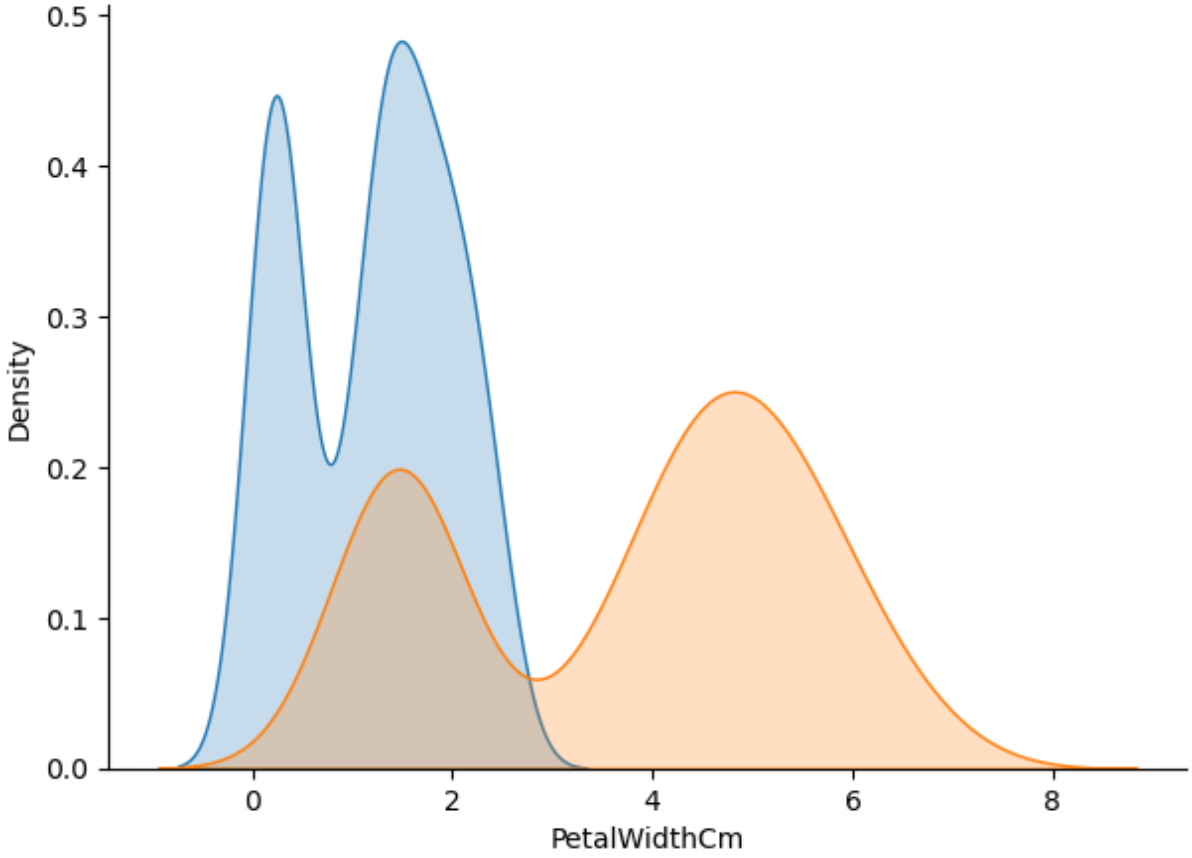
Out[19]:
```
<AxesSubplot:xlabel='Density', ylabel='PetalWidthCm'>
```



Multi Variable Density Plot

In [20]:
```
f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)
sns.kdeplot(iris['PetalWidthCm'], shade=True)
sns.kdeplot(iris['PetalLengthCm'], shade=True)
plt.show()
```

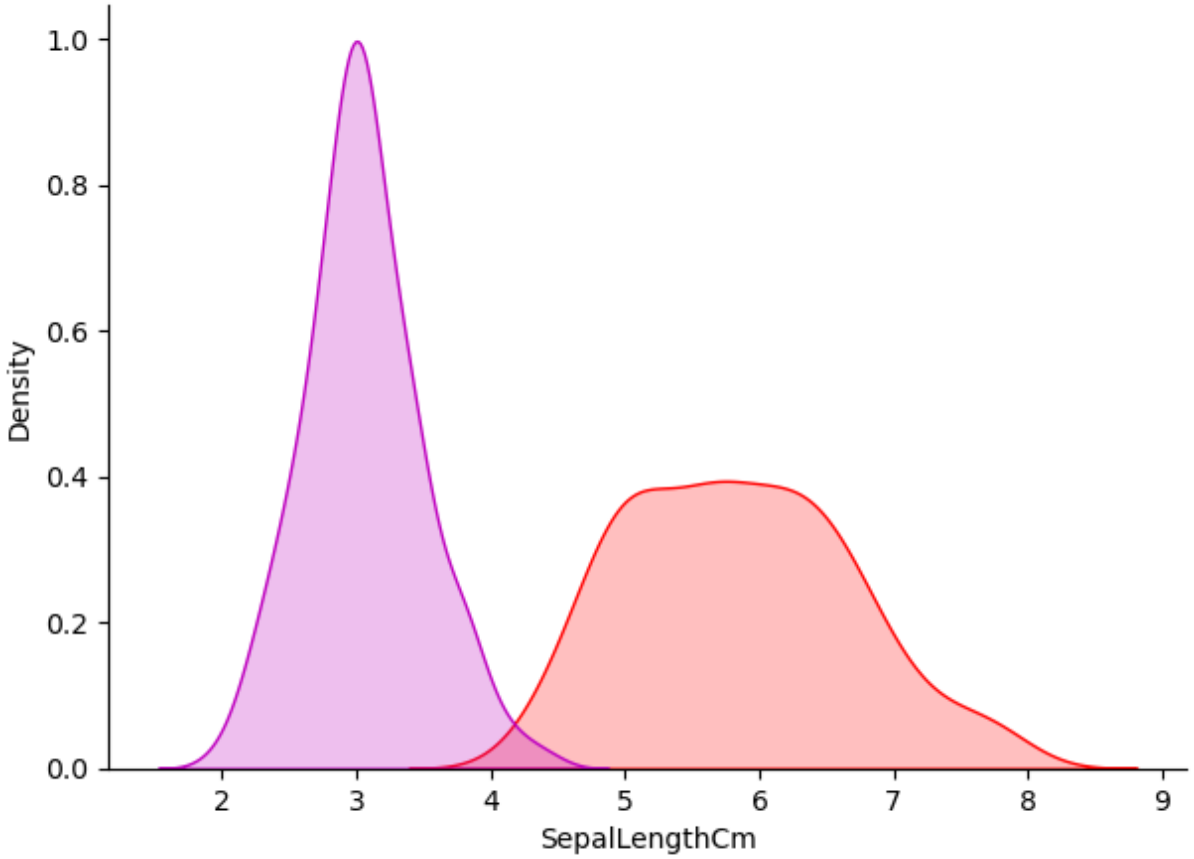
```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  This is separate from the ipykernel package so we can avoid doing imports until
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:4: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  after removing the cwd from sys.path.
```



Multi Variable Density Plot: Customize Color

In [21]:
```
f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)
sns.kdeplot(iris['SepalLengthCm'], shade=True, color='r')
sns.kdeplot(iris['SepalWidthCm'], shade=True, color='m')
plt.show()
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  This is separate from the ipykernel package so we can avoid doing imports until
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:4: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  after removing the cwd from sys.path.
```

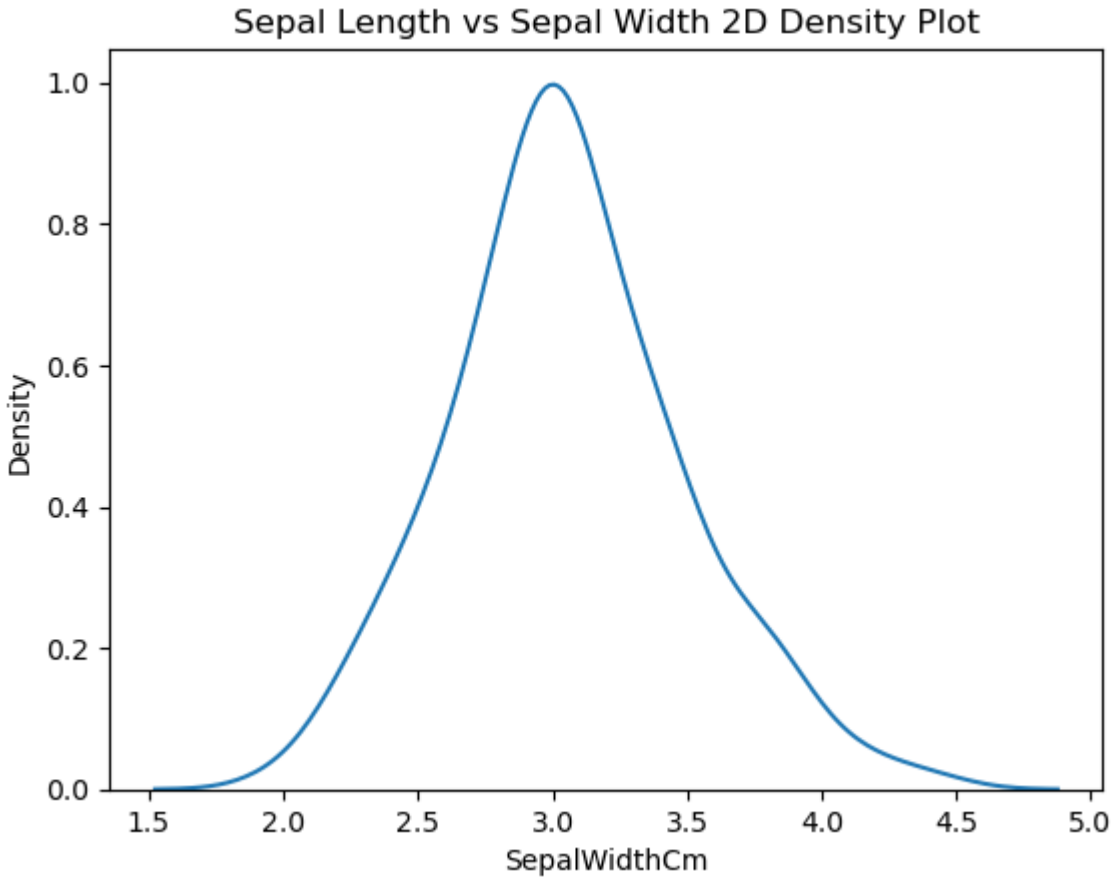


2D Density Plot

Basic 2D Density Plot

In [22]:
```
sns.kdeplot(data = iris, x = 'SepalWidthCm')
plt.title("Sepal Length vs Sepal Width 2D Density Plot")
```

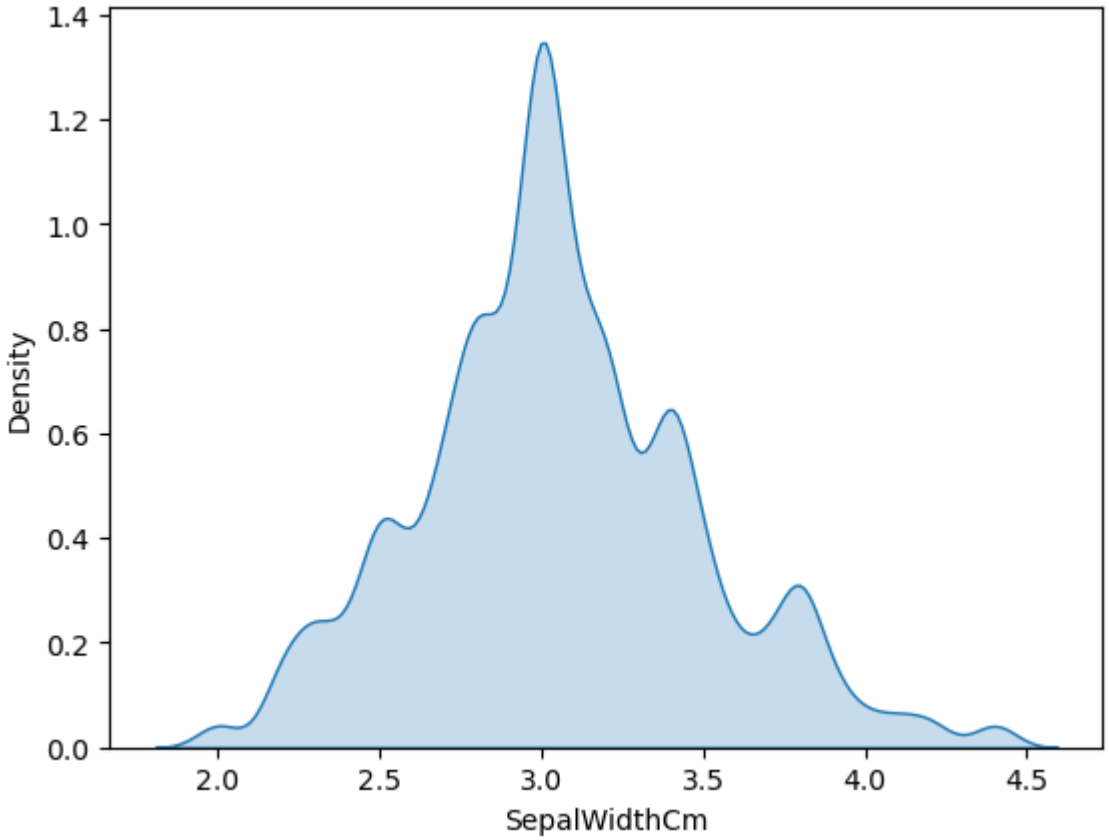
Out[22]:
```
Text(0.5, 1.0, 'Sepal Length vs Sepal Width 2D Density Plot')
```



In [23]:
```
sns.kdeplot(data = iris, x = 'SepalWidthCm', cmap="Reds", shade=True, bw=.15)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and `bw_adjust`.
Setting `bw_method=0.15`, but please see the docs for the new parameters
and update your code. This will become an error in seaborn v0.13.0.

  """Entry point for launching an IPython kernel.
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  """Entry point for launching an IPython kernel.
```

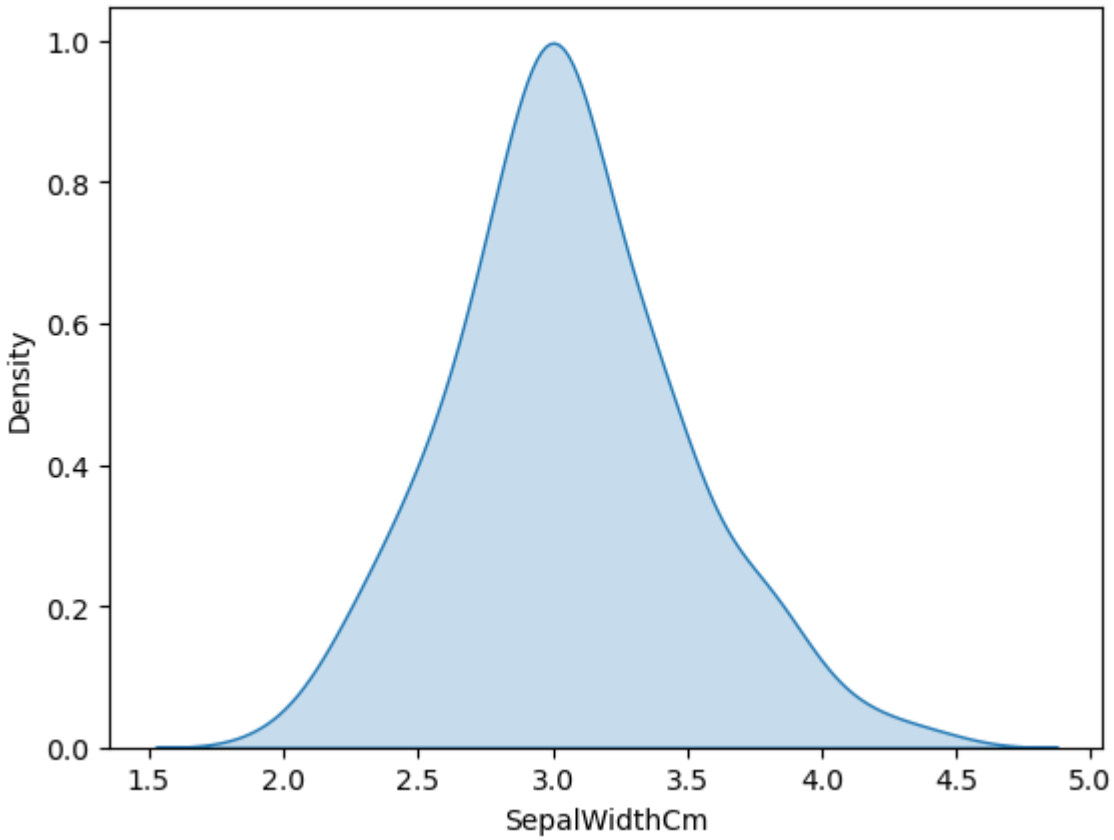
Out[23]:
```
<AxesSubplot:xlabel='SepalWidthCm', ylabel='Density'>
```



In [24]:
```
sns.kdeplot(data = iris, x = 'SepalWidthCm', cmap="Blues", shade=True, shade_lowest=
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: UserWarning:

`shade_lowest` has been replaced by `thresh`; setting `thresh=0.
This will become an error in seaborn v0.13.0; please update your code.

  """Entry point for launching an IPython kernel.
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  """Entry point for launching an IPython kernel.
```

Out[24]:
```
<AxesSubplot:xlabel='SepalWidthCm', ylabel='Density'>
```

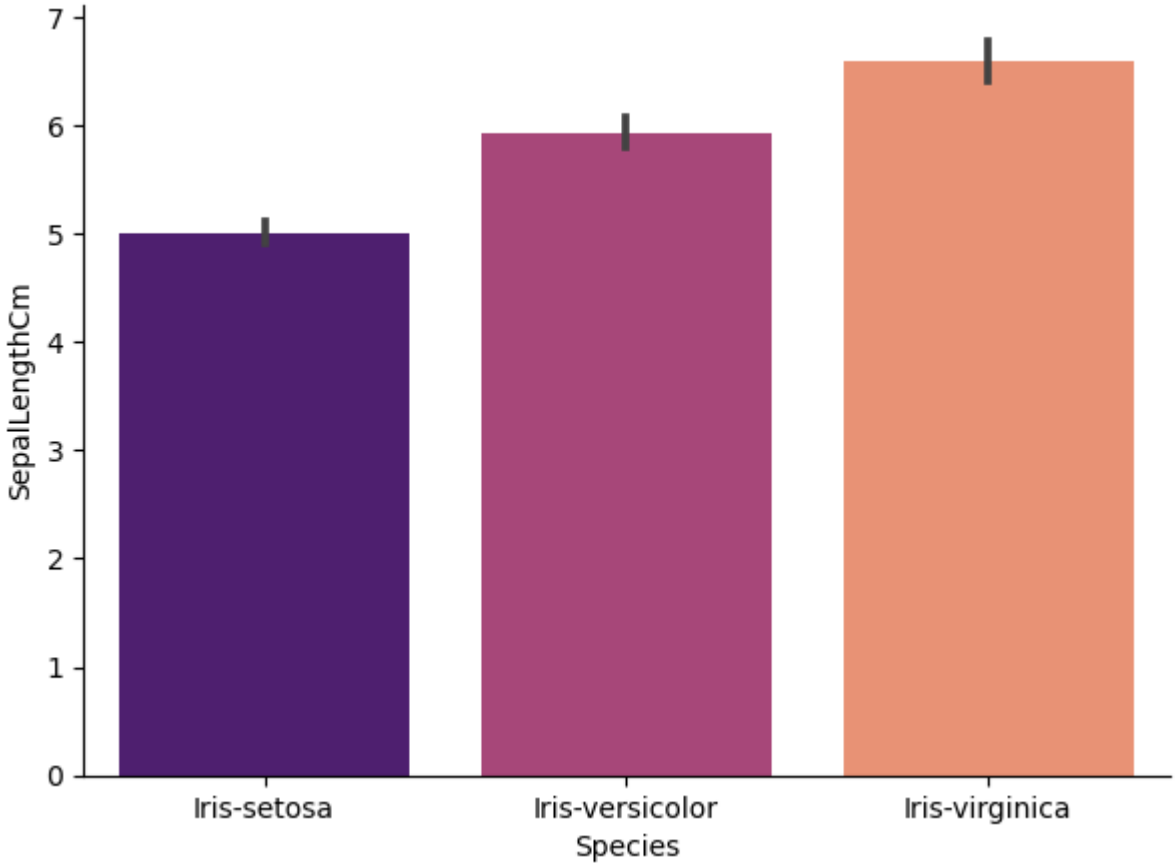


Bar Plot

Basic Bar Plot

In [25]:
```
f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)
sns.barplot(x='Species', y='SepalLengthCm', data=iris, palette='magma')
```

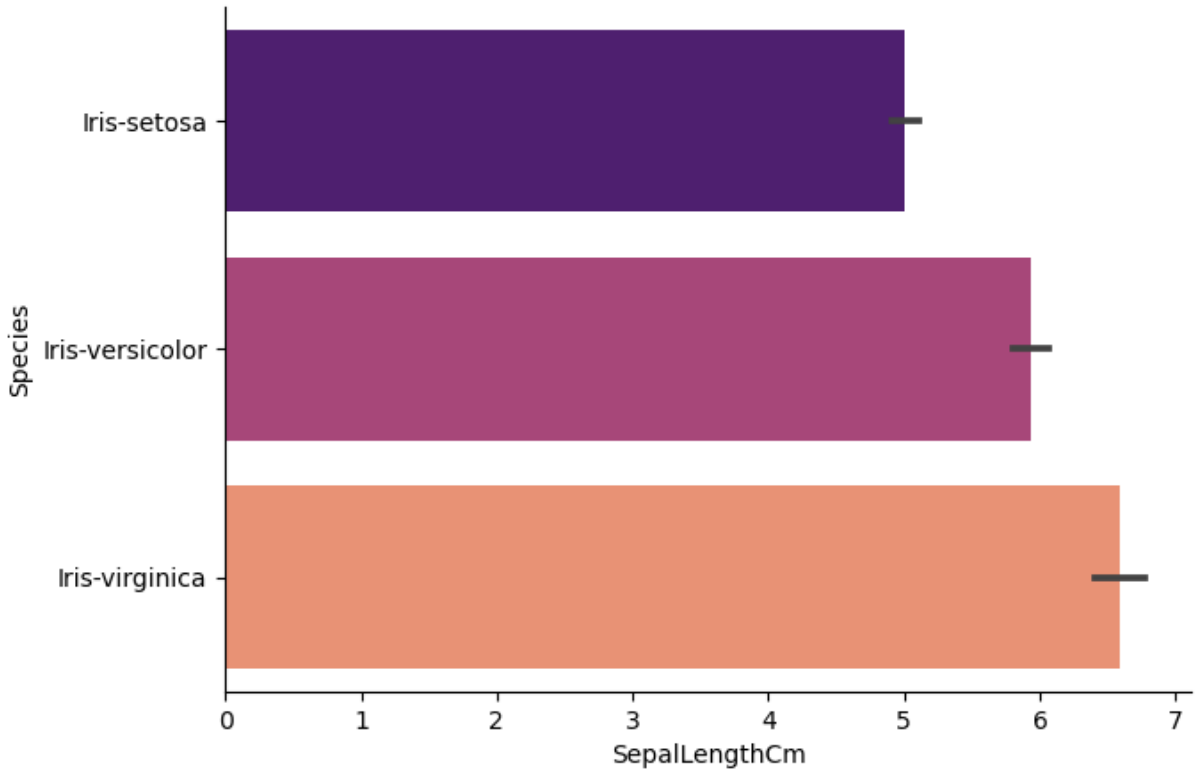
Out[25]:
```
<AxesSubplot:xlabel='Species', ylabel='SepalLengthCm'>
```



Horizontal Bar Plot

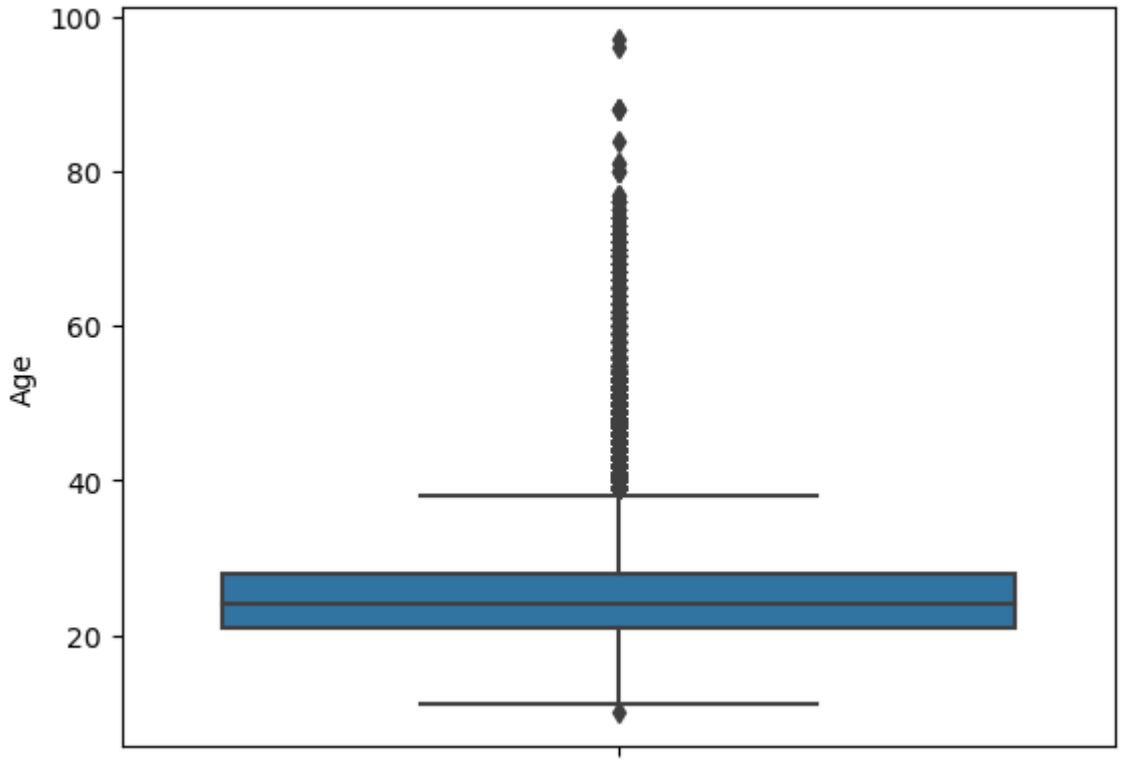
In [26]:
```
f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)
sns.barplot(x='SepalLengthCm', y='Species', data=iris, palette='magma', orient='h')
```

Out[26]:
```
<AxesSubplot:xlabel='SepalLengthCm', ylabel='Species'>
```



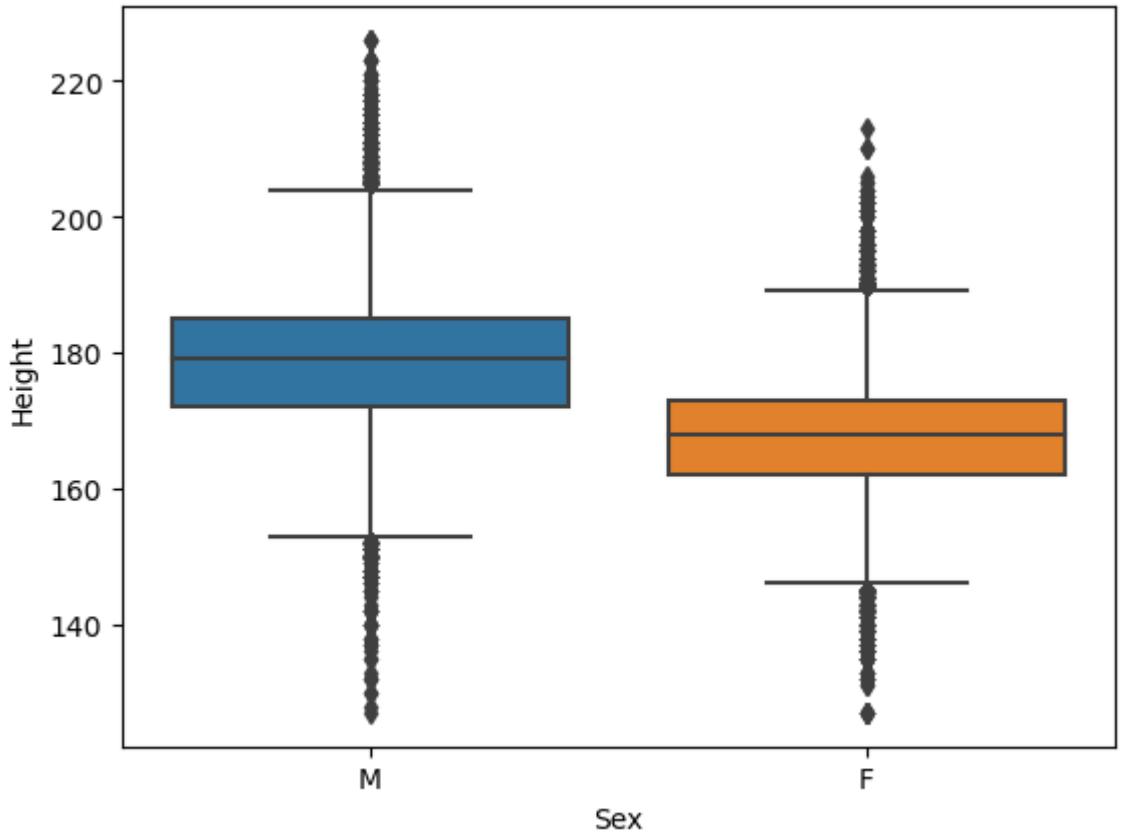
Box Plot

In [27]:
```
sns.boxplot(y=olympic['Age'])
plt.show()
```



Box Plot - Category

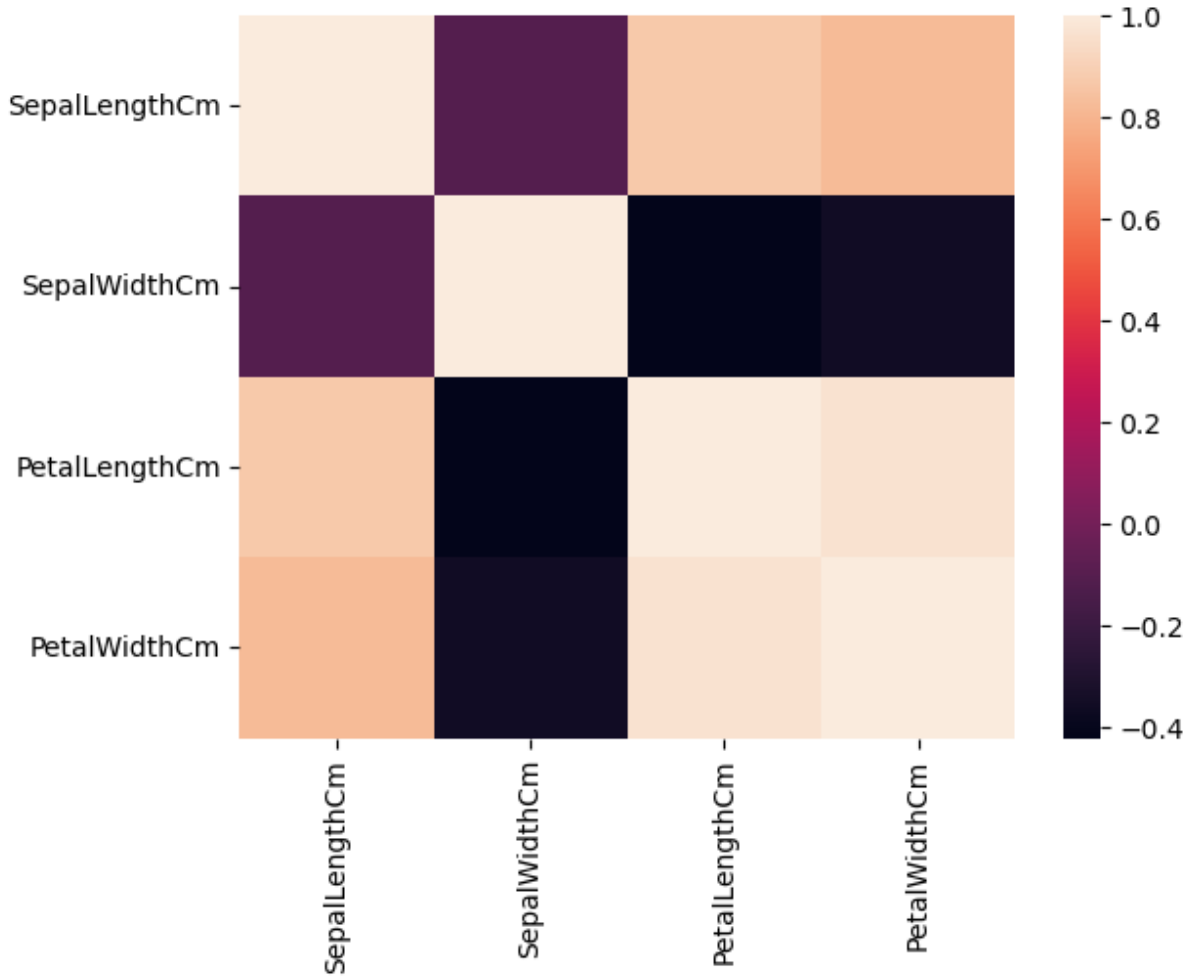
In [28]:
```
sns.boxplot(x=olympic['Sex'], y=olympic['Height'])
plt.show()
```



Heatmap

In [29]:
```
sns.heatmap(iris.drop(['Id'], axis=1).corr())
```

Out[29]:
```
<AxesSubplot:>
```

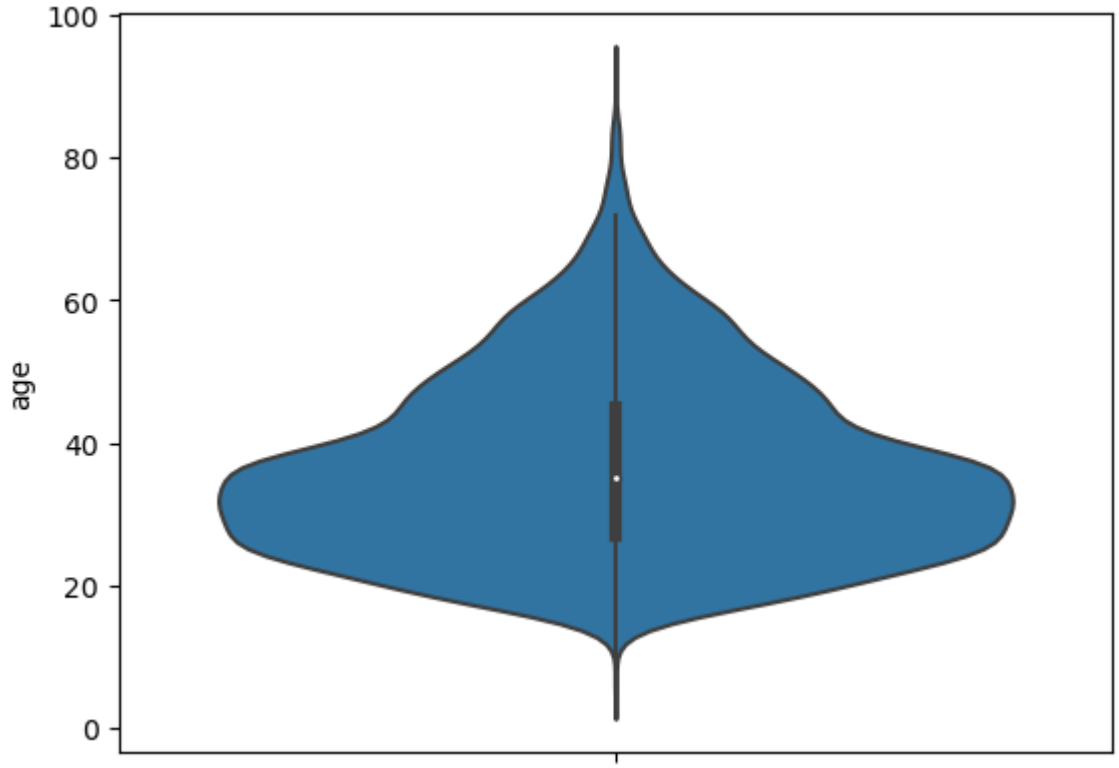


Violin Plot

Basic Violin Plot

In [30]:
```
sns.violinplot(y=shootings['age'])
```

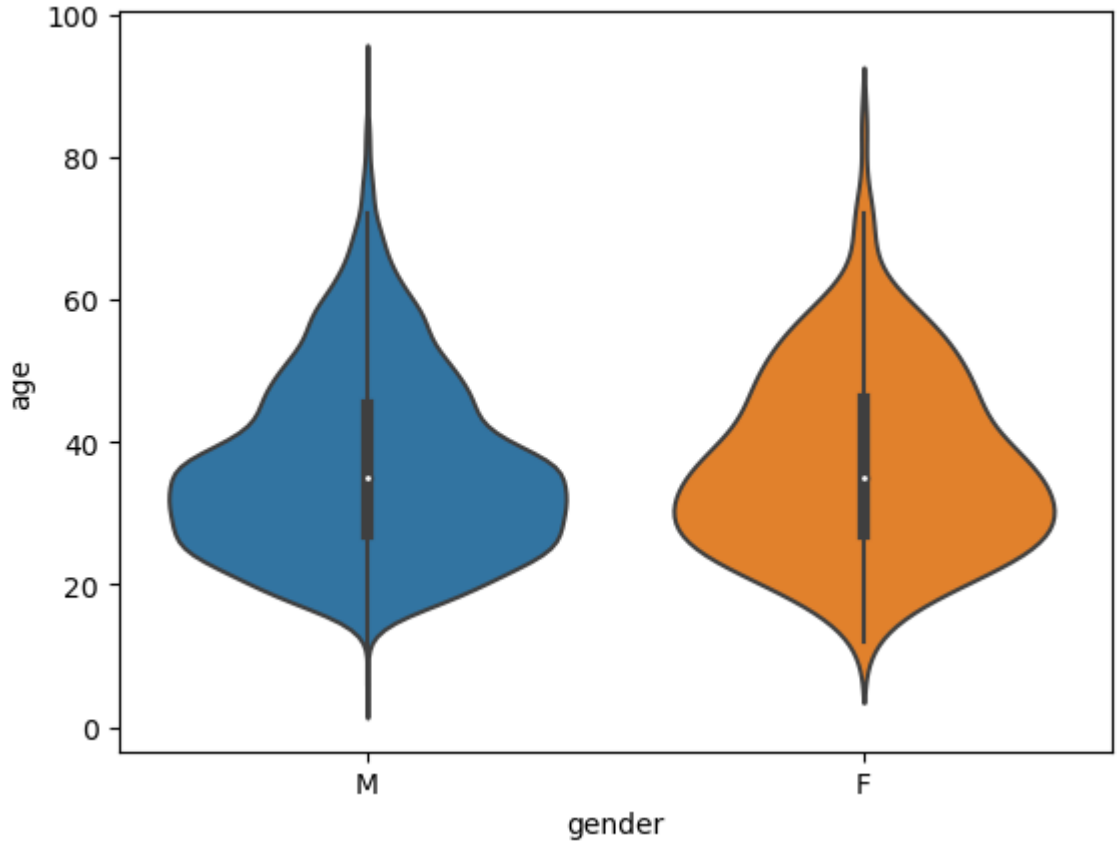
Out[30]:
```
<AxesSubplot:ylabel='age'>
```



Violin Plot - Category

In [31]:
```
sns.violinplot(x=shootings['gender'], y=shootings['age'])
```

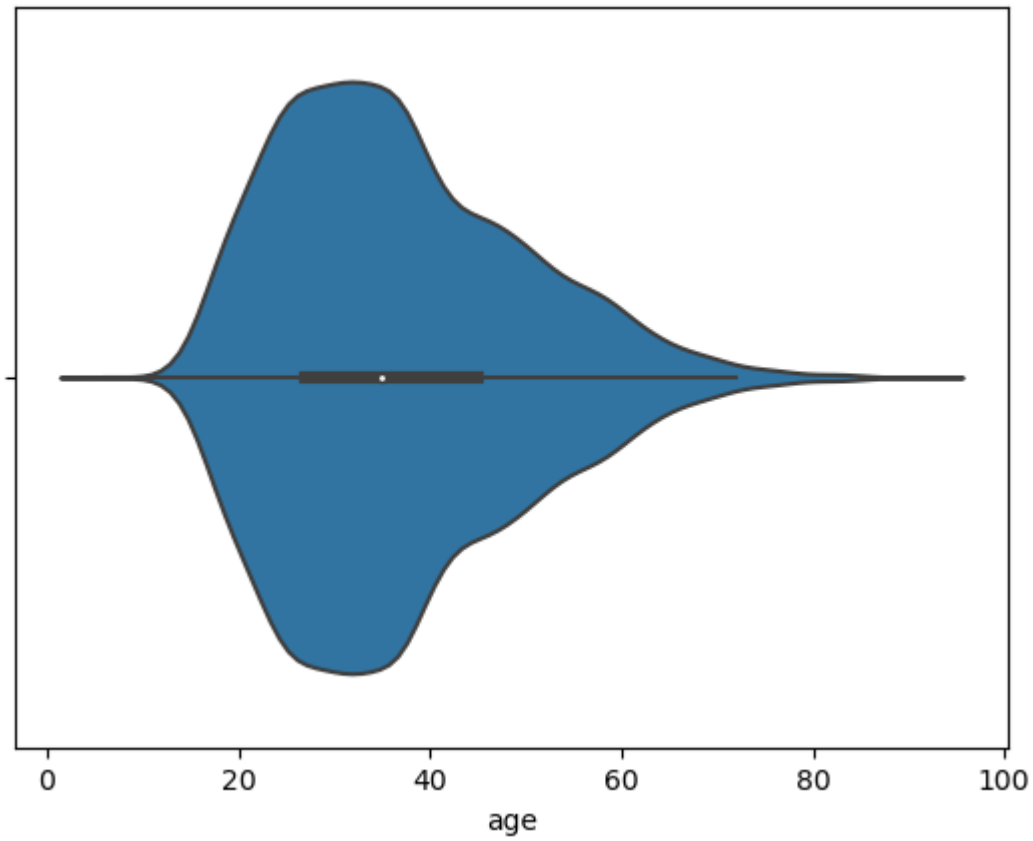
Out[31]:
```
<AxesSubplot:xlabel='gender', ylabel='age'>
```



Horizontal Violin Plot

In [32]:
```
sns.violinplot(x=shootings['age'])
```

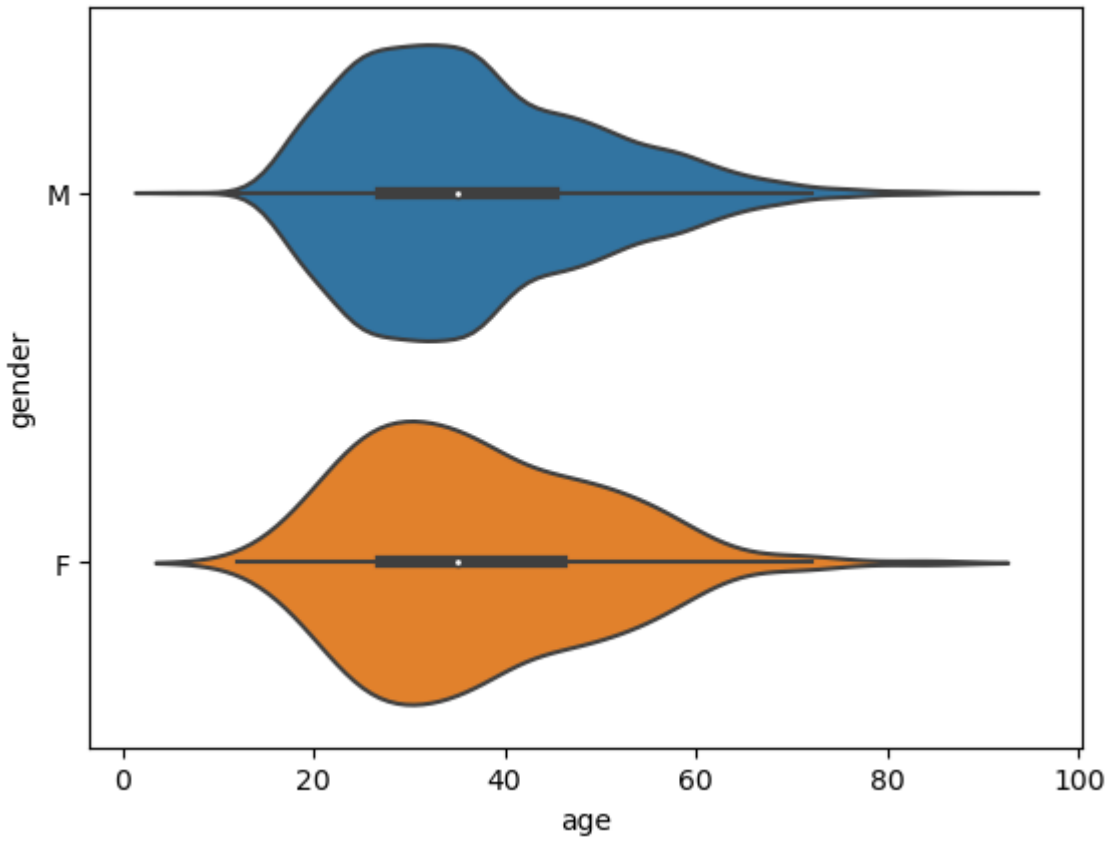
Out[32]:
```
<AxesSubplot:xlabel='age'>
```



Horizontal Violin Plot - Category

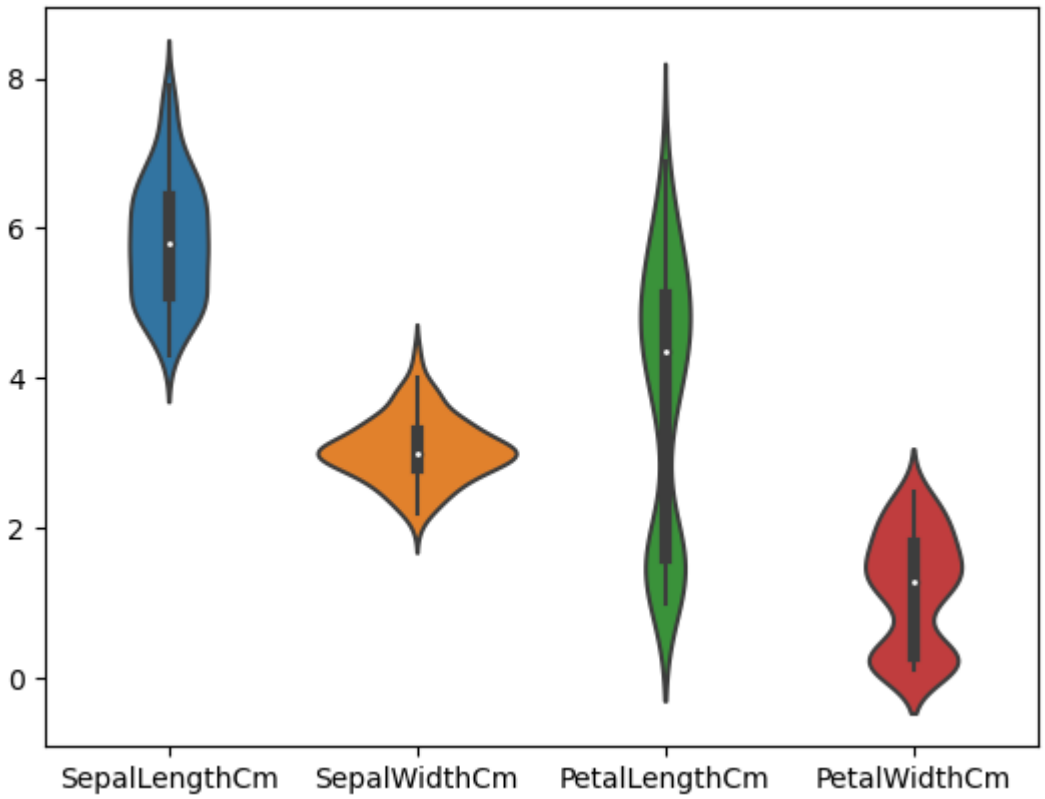
In [33]:
```
sns.violinplot(y=shootings['gender'], x=shootings['age'])
```

Out[33]:
```
<AxesSubplot:xlabel='age', ylabel='gender'>
```



In [34]:
```
sns.violinplot(data=iris.iloc[:,1:5])
```

Out[34]: <AxesSubplot:>



Seaborn Themes

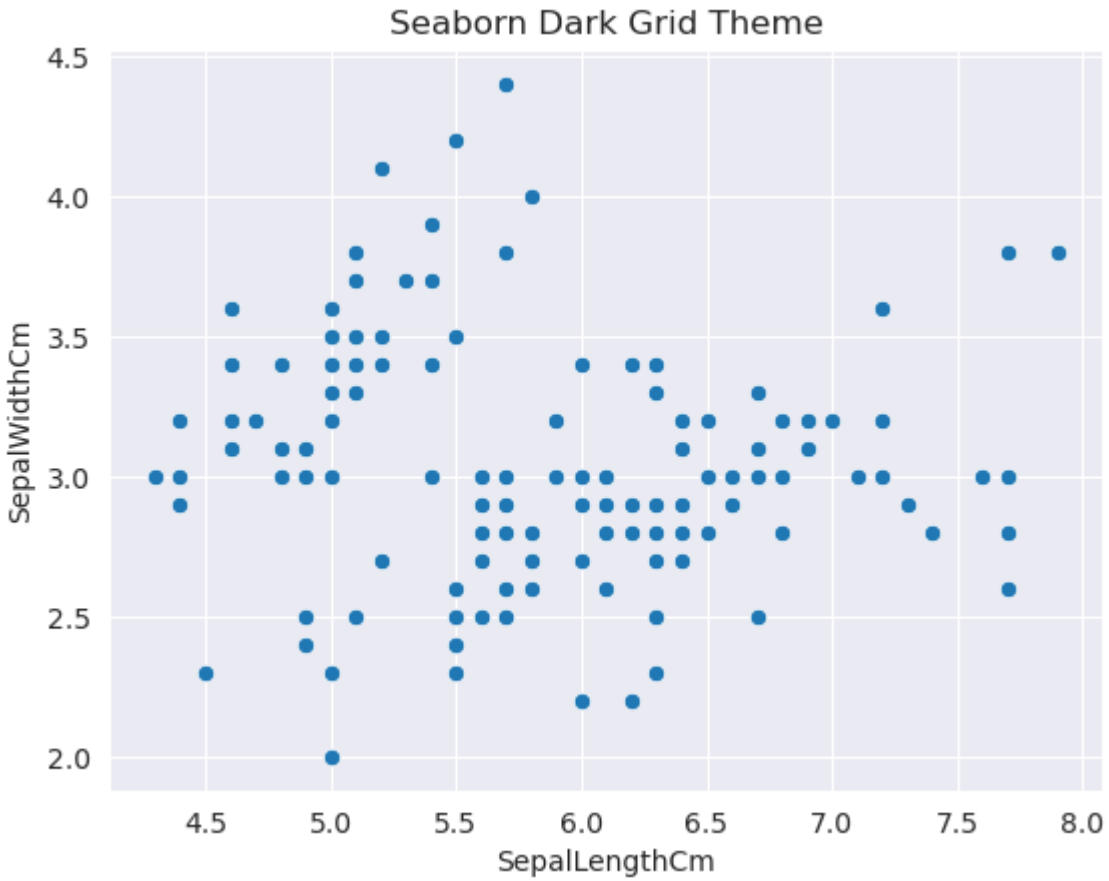
Seaborn has five built-in themes to style its plots: `darkgrid`, `whitegrid`, `dark`, `white`, and `ticks`. Seaborn defaults to using the darkgrid theme for its plots, but you can change this styling to better suit your presentation needs.

To use any of the preset themes pass the name of it to `sns.set_style()`.

Seaborn Dark Grid Theme

In [35]:
```
sns.set_style("darkgrid")
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm', data=iris)
plt.title("Seaborn Dark Grid Theme")
```

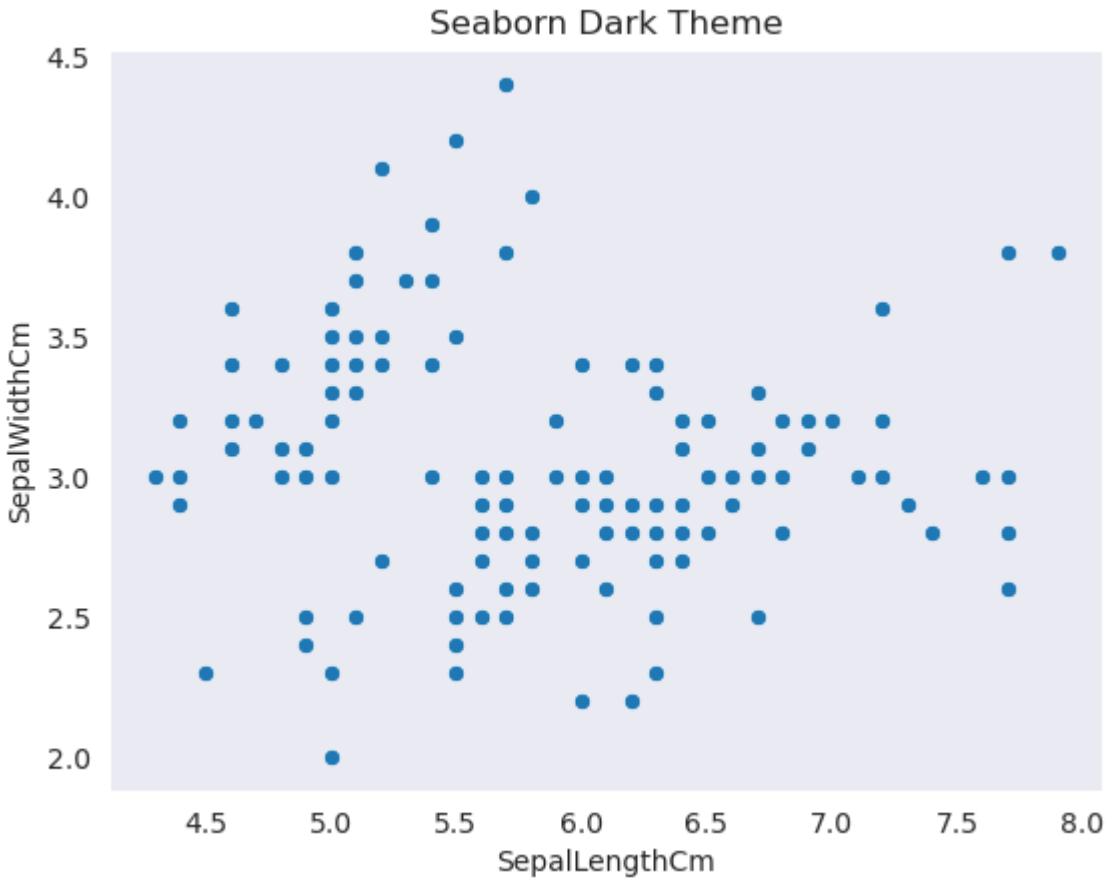
Out[35]:
```
Text(0.5, 1.0, 'Seaborn Dark Grid Theme')
```



Seaborn Dark Theme

In [36]:
```
sns.set_style("dark")
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm', data=iris)
plt.title("Seaborn Dark Theme")
```

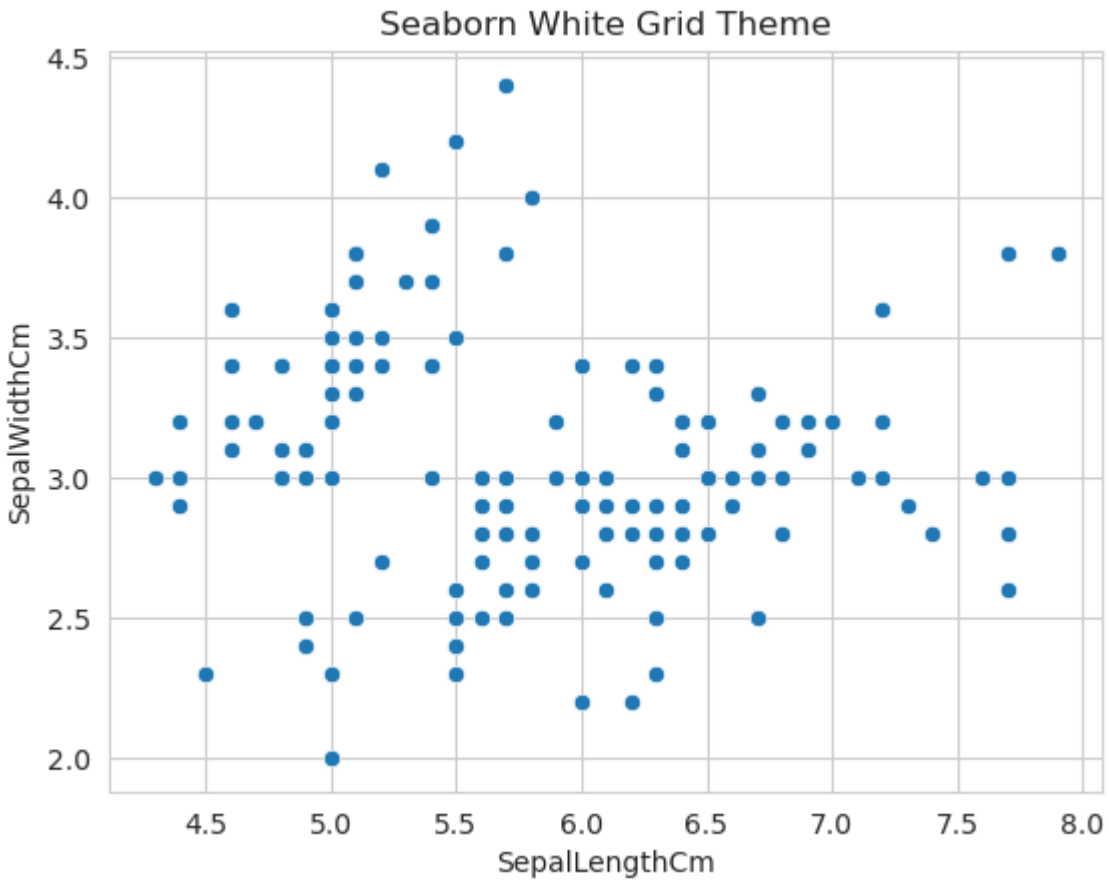
Out[36]:
```
Text(0.5, 1.0, 'Seaborn Dark Theme')
```



Seaborn White Grid Theme

In [37]:
```
sns.set_style("whitegrid")
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm', data=iris)
plt.title("Seaborn White Grid Theme")
```

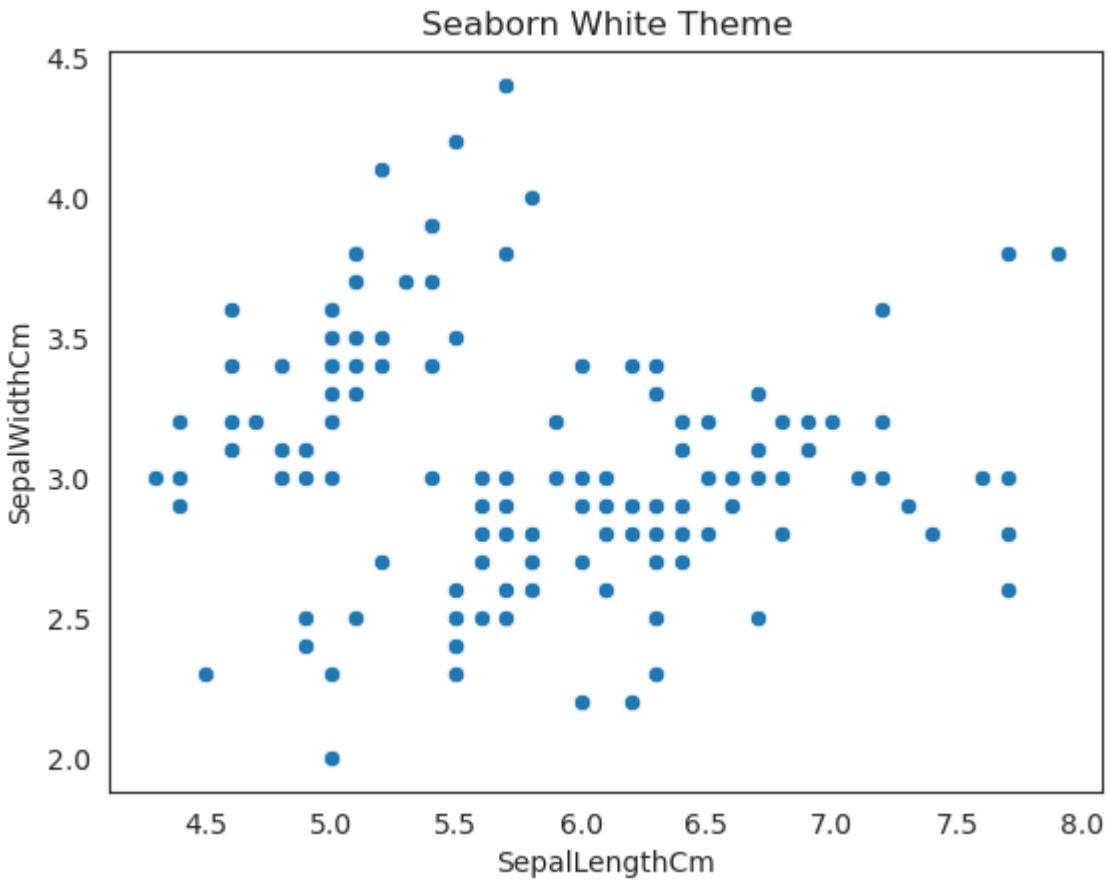
Out[37]:
```
Text(0.5, 1.0, 'Seaborn White Grid Theme')
```



Seaborn White Theme

In [38]:
```
sns.set_style("white")
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm', data=iris)
plt.title("Seaborn White Theme")
```

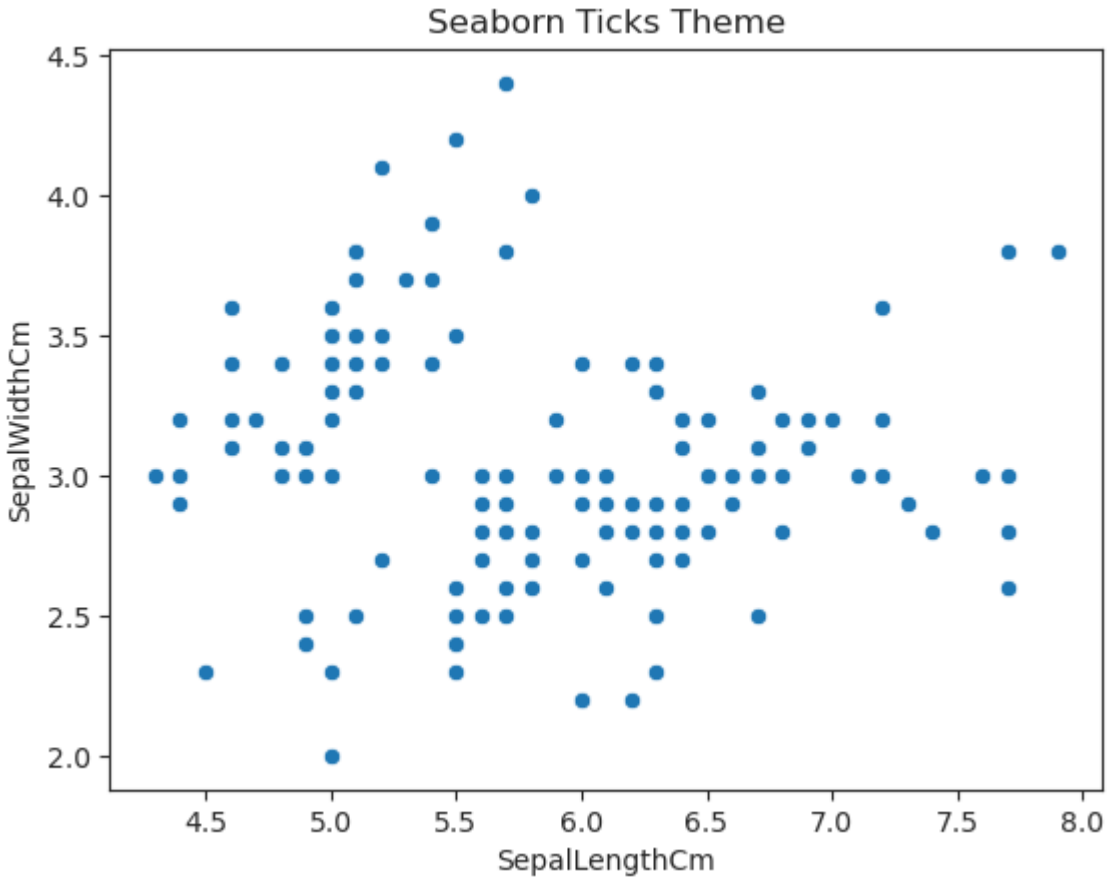
Out[38]:
```
Text(0.5, 1.0, 'Seaborn White Theme')
```



Seaborn Ticks Theme

In [39]:
```
sns.set_style("ticks")
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm', data=iris)
plt.title("Seaborn Ticks Theme")
```

Out[39]:
```
Text(0.5, 1.0, 'Seaborn Ticks Theme')
```



Seaborn is a huge library. Of course, these are not all. But this article covers a lot! There is a lot more to learn about this library.

Work in Progress... ⏳

Ref: https://www.python-graph-gallery.com/cheat-sheets/

Ref: https://regenerativetoday.com/an-ultimate-cheat-sheet-for-stylish-data-visualization-in-pythons-seaborn-library/

Ref: https://www.paradigmadigital.com/wp-content/uploads/2019/06/cheat-sheet-pandas-seaborn.pdf

In [ ]: