In [1]:
```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

import pandas as pd
pd.plotting.register_matplotlib_converters()
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
fifa_data = pd.read_csv("../Data/fifa.csv", index_col = "Date", parse_dates = Tr
```

In [3]:
```
fifa_data.head()
```

Out[3]:

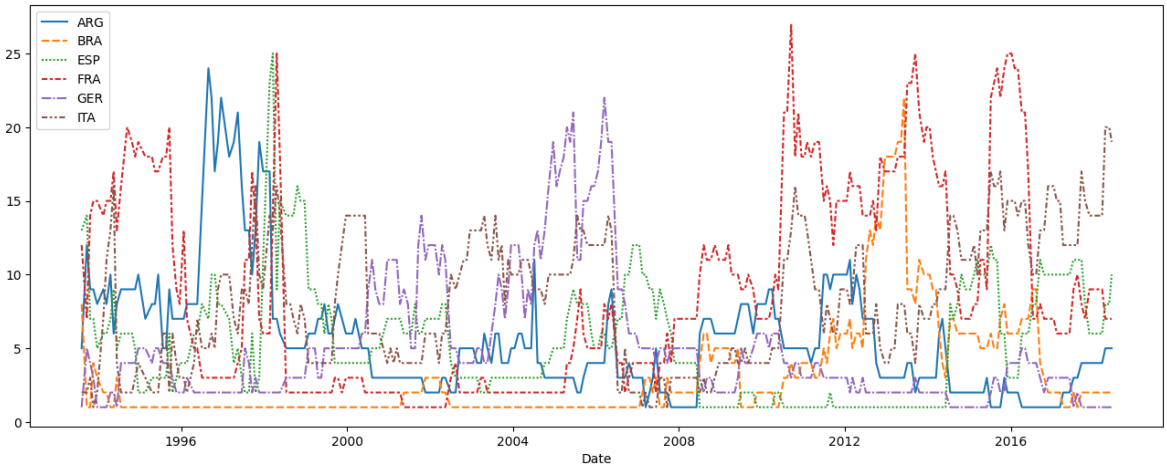
Date	ARG	BRA	ESP	FRA	GER	ITA
1993-08-08	5.0	8.0	13.0	12.0	1.0	2.0
1993-09-23	12.0	1.0	14.0	7.0	5.0	2.0
1993-10-22	9.0	1.0	7.0	14.0	4.0	3.0
1993-11-19	9.0	4.0	7.0	15.0	3.0	1.0
1993-12-23	8.0	3.0	5.0	15.0	1.0	2.0

Plot

In [4]:
```
# Set the width and height of the figure
plt.figure(figsize=(16,6))

# Line chart showing how FIFA rankings evolved over time
sns.lineplot(data=fifa_data)
```

Out[4]:
```
<Axes: xlabel='Date'>
```



Spotify

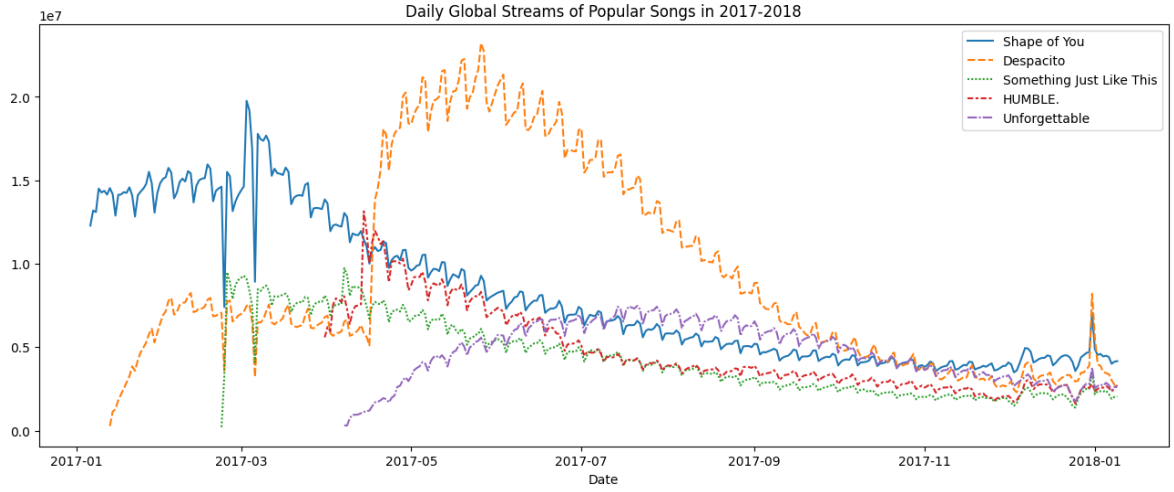
In [5]:
```
db = pd.read_csv("../Data/spotify.csv", index_col = "Date", parse_dates = True)
db.head()
```

Out[5]:

Date	Shape of You	Despacito	Something Just Like This	HUMBLE.	Unforgettable
2017-01-06	12287078	NaN	NaN	NaN	NaN
2017-01-07	13190270	NaN	NaN	NaN	NaN
2017-01-08	13099919	NaN	NaN	NaN	NaN
2017-01-09	14506351	NaN	NaN	NaN	NaN
2017-01-10	14275628	NaN	NaN	NaN	NaN

In [6]:
```
plt.figure(figsize=(16,6))
plt.title("Daily Global Streams of Popular Songs in 2017-2018")
sns.lineplot(data = db)
```

Out[6]:
```
<Axes: title={'center': 'Daily Global Streams of Popular Songs in 2017-2018'},
xlabel='Date'>
```



Plot A Subset Of Data

In [7]:
```
list(db.columns)
```

Out[7]:
```
['Shape of You',
 'Despacito',
 'Something Just Like This',
 'HUMBLE.',
 'Unforgettable']
```

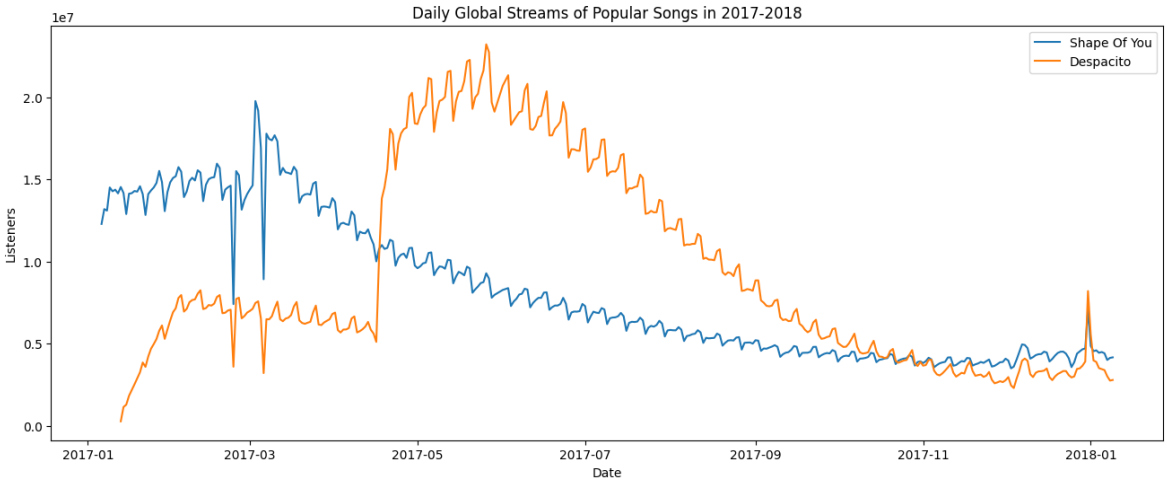
In [8]:
```
plt.figure(figsize=(16,6))

plt.title("Daily Global Streams of Popular Songs in 2017-2018")

sns.lineplot(data = db["Shape of You"], label = "Shape Of You")
sns.lineplot(data = db["Despacito"], label = "Despacito")

plt.xlabel("Date")
plt.ylabel("Listeners")
```

Out[8]:
```
Text(0, 0.5, 'Listeners')
```



Bar Charts

In [9]:
```
import pandas as pd
pd.plotting.register_matplotlib_converters()
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
print("Setup Complete")
```

```
Setup Complete
```

In [10]:
```
db = pd.read_csv("../Data/flight_delays.csv", index_col = 'Month')
db.head()
```

Out[10]:

Month	AA	AS	B6	DL	EV	F9	HA	
1	6.955843	-0.320888	7.347281	-2.043847	8.537497	18.357238	3.512640	18.16
2	7.530204	-0.782923	18.657673	5.614745	10.417236	27.424179	6.029967	21.30
3	6.693587	-0.544731	10.741317	2.077965	6.730101	20.074855	3.468383	11.01
4	4.931778	-3.009003	2.780105	0.083343	4.821253	12.640440	0.011022	5.13
5	5.173878	-1.716398	-0.709019	0.149333	7.724290	13.007554	0.826426	5.46

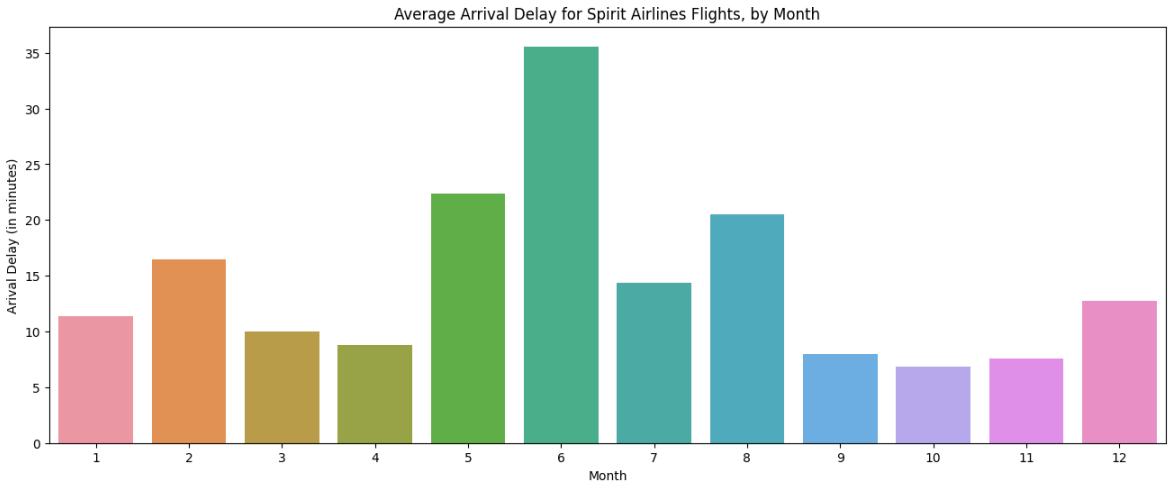
In [11]:
```
db["US"] = db["US"].fillna(db["US"].mean(skipna= True))
```

In [12]:
```
plt.figure(figsize=(16,6))
plt.title("Average Arrival Delay for Spirit Airlines Flights, by Month")

sns.barplot(x = db.index, y = db['NK'])

plt.ylabel("Arival Delay (in minutes)")
```

Out[12]: Text(0, 0.5, 'Arival Delay (in minutes)')

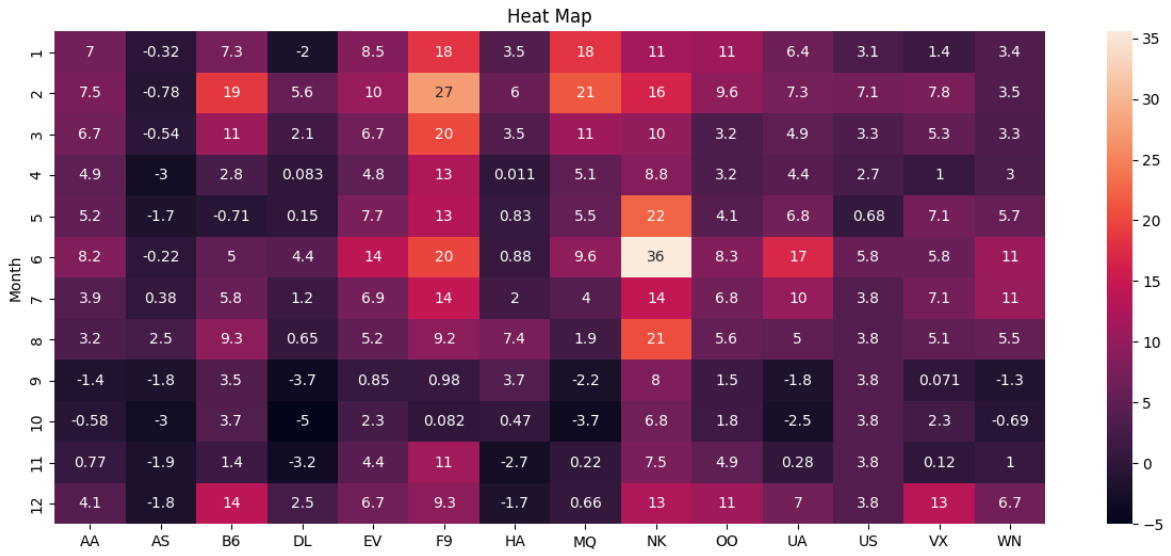


Heat Map

In [13]:
```
plt.figure(figsize=(15,6))
plt.title("Heat Map")

sns.heatmap(db,annot = True)
#annot for annotation = notlarla açıklama
```

Out[13]: <Axes: title={'center': 'Heat Map'}, ylabel='Month'>



We'll work with a (synthetic) dataset of insurance charges, to see if we can understand why some customers pay more than others.

In [14]:
```
db = pd.read_csv("../Data/insurance.csv")
db.head()
```

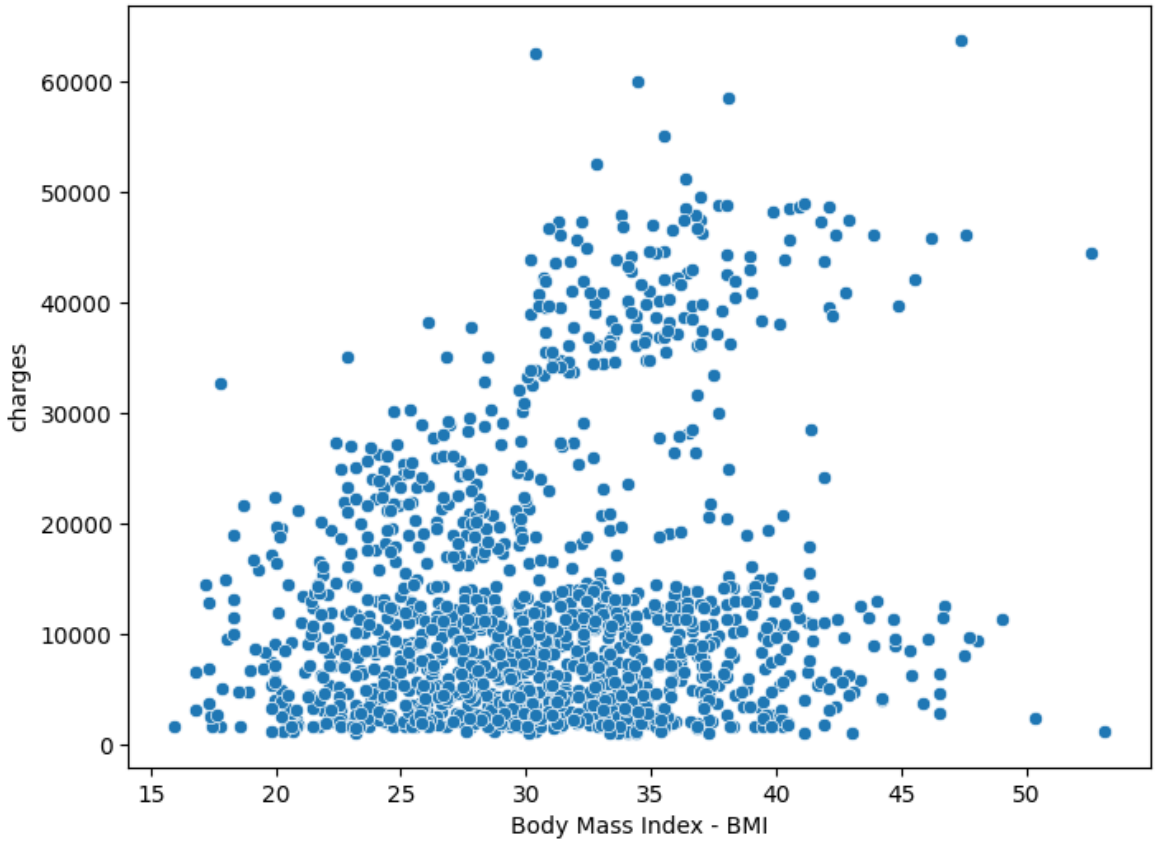
Out[14]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

Scatter

In [15]:
```
plt.figure(figsize= (8,6))
plt.xlabel("Body Mass Index - BMI")
sns.scatterplot(x = db["bmi"], y = db["charges"])
```

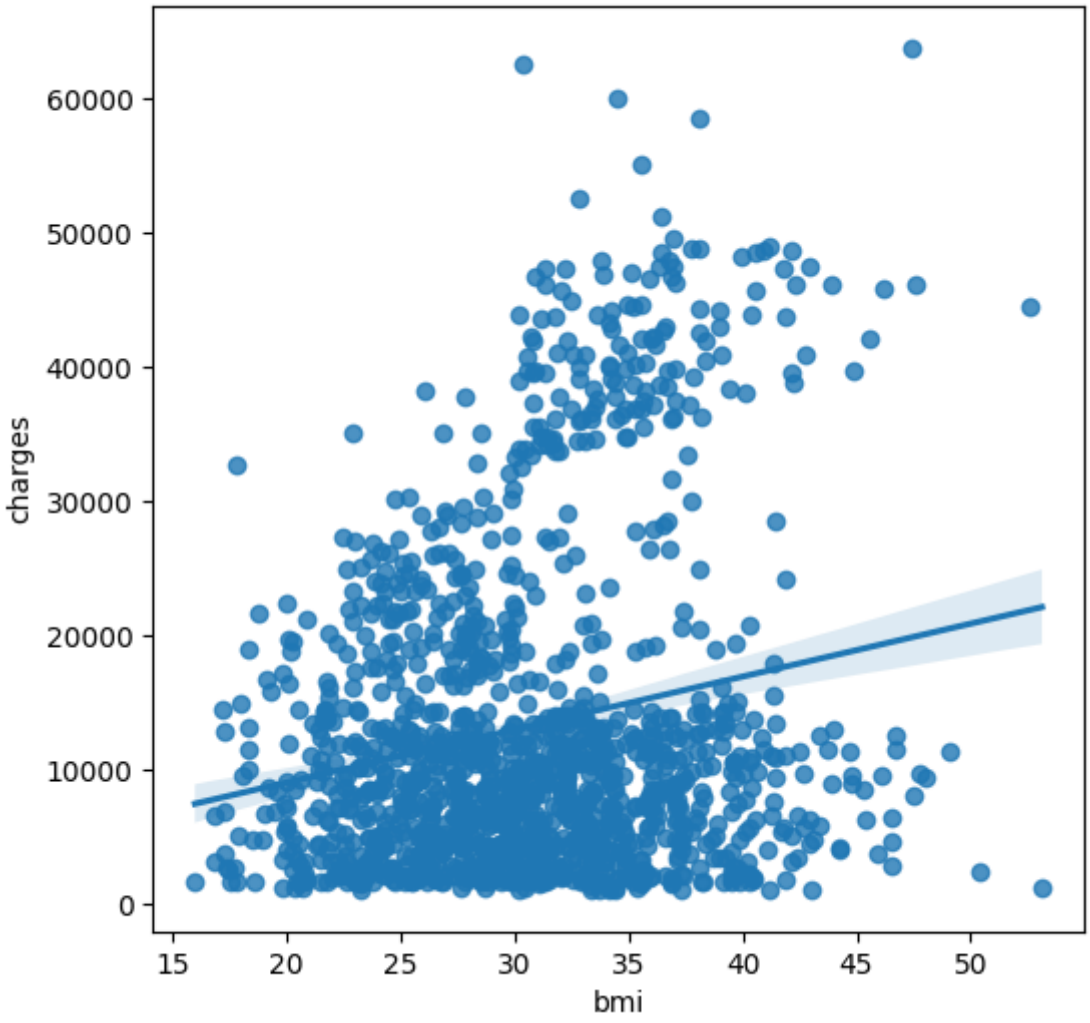
Out[15]:
```
<Axes: xlabel='Body Mass Index - BMI', ylabel='charges'>
```



Regression Line

In [16]:
```
plt.figure(figsize=(6,6))
sns.regplot(x = db["bmi"], y = db["charges"])
```

Out[16]:
```
<Axes: xlabel='bmi', ylabel='charges'>
```

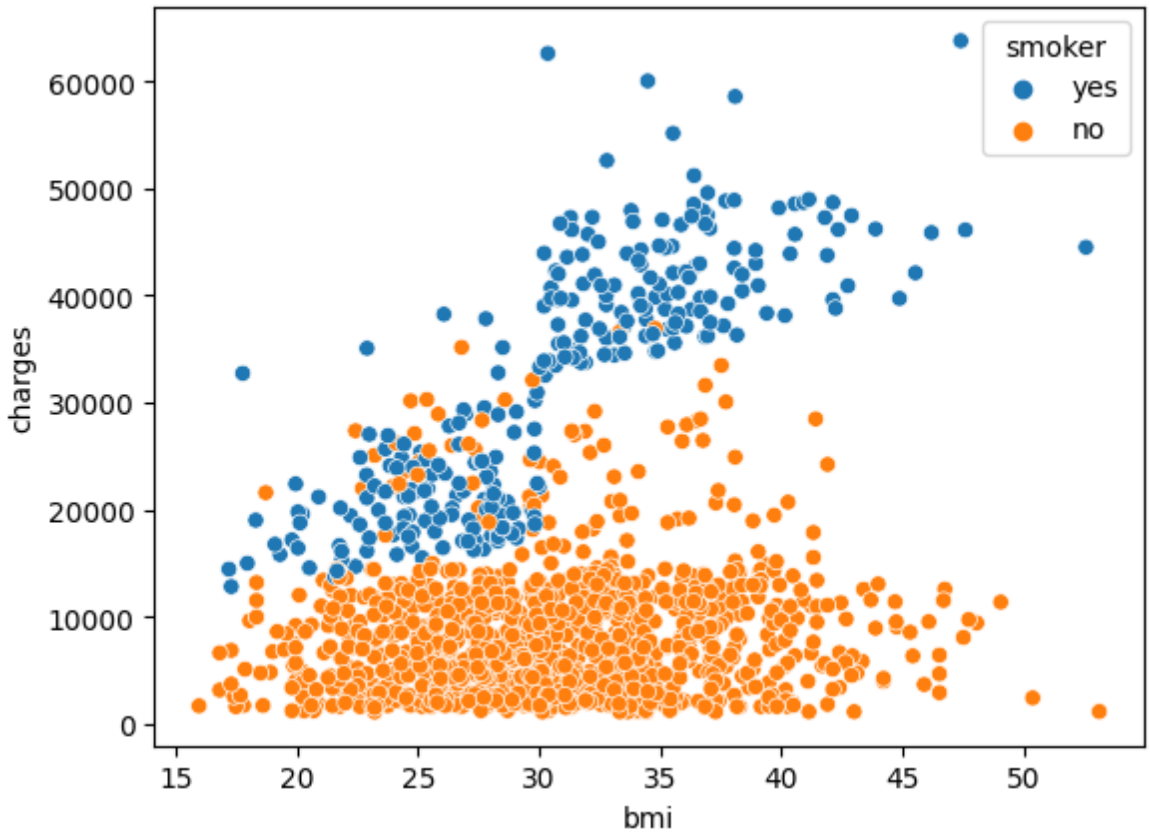


Color Coded

We can use scatter plots to display the relationships between (not two, but...) three variables! One way of doing this is by color-coding the points.

In [17]:
```
sns.scatterplot(x=db["bmi"], y= db["charges"], hue=db["smoker"])
```

Out[17]:
```
<Axes: xlabel='bmi', ylabel='charges'>
```



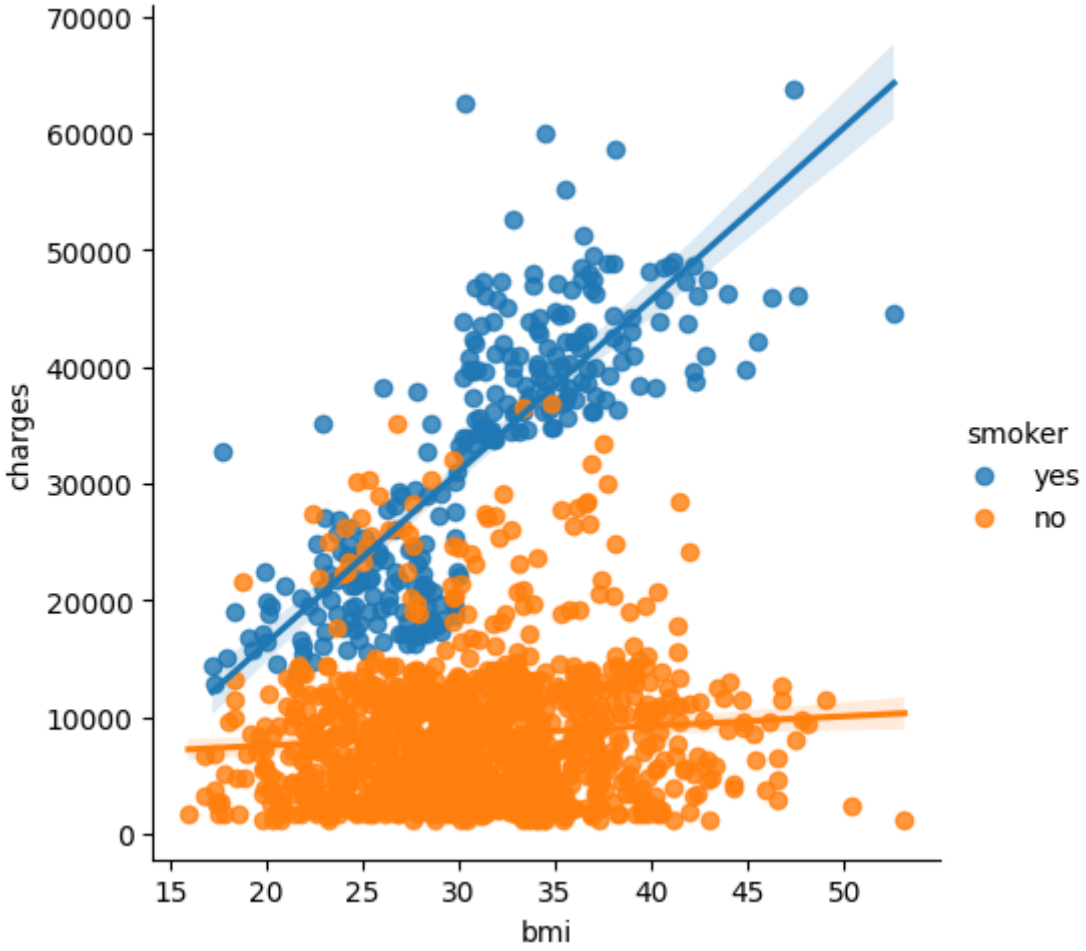
lmplot - linear model plot

In [18]:
```
plt.figure(figsize=(8,8))
sns.lmplot(x = "bmi", y = "charges", hue = "smoker", data = db)
```

```
C:\Users\ulasu\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[18]:
```
<seaborn.axisgrid.FacetGrid at 0x1ff5dd5b3d0>
```

```
<Figure size 800x800 with 0 Axes>
```



- Instead of setting `x=insurance_data['bmi']` to select the `'bmi'` column in `insurance_data`, we set `x="bmi"` to specify the name of the column only.
- Similarly, `y="charges` " and `hue="smoker` " also contain the names of columns
- We specify the dataset with data=insurance_data.

Categorical Scatter Plot

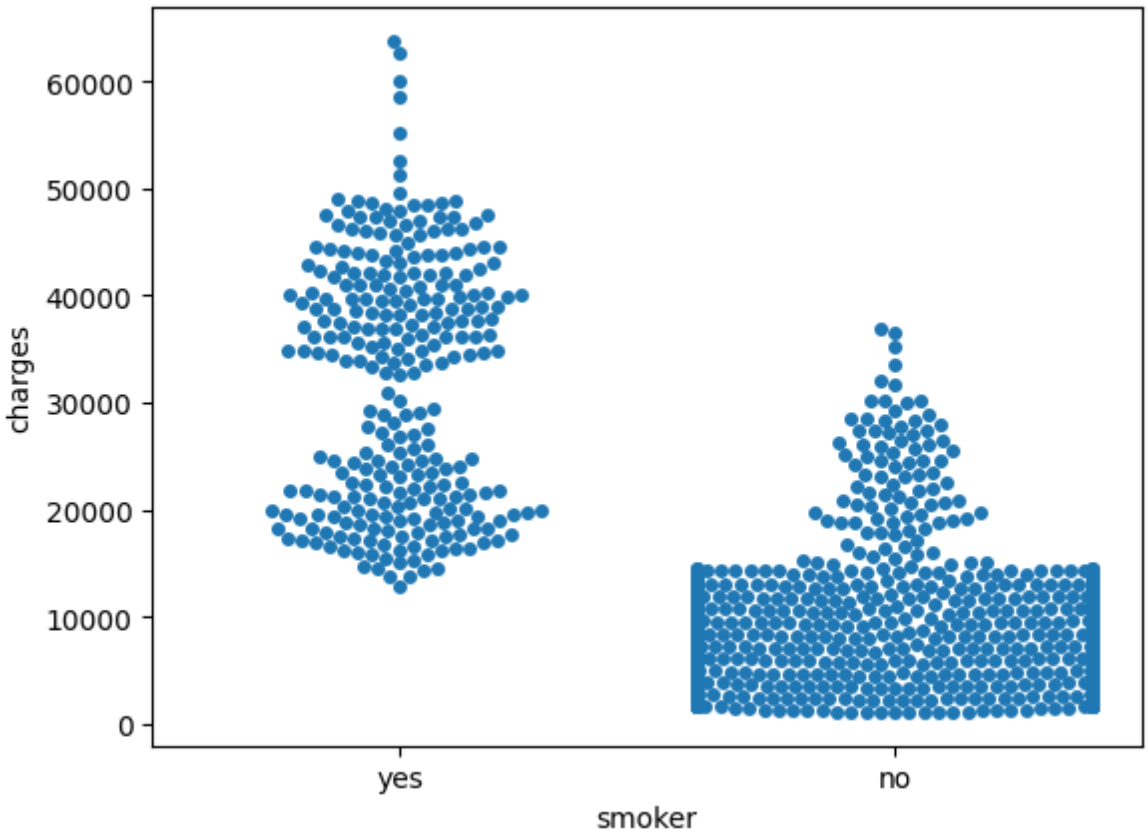
swarmplot

In [19]:
```
sns.swarmplot(x = db["smoker"], y = db["charges"])
```

```
C:\Users\ulasu\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\categorical.py:3544: UserWarning: 37.4% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[19]:
```
<Axes: xlabel='smoker', ylabel='charges'>
```

```
C:\Users\ulasu\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\categorical.py:3544: UserWarning: 60.8% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



Each row in the dataset corresponds to a different flower. There are four measurements: the sepal length and width, along with the petal length and width. We also keep track of the corresponding species.

In [20]:
```
db = pd.read_csv("../Data/iris.csv", index_col = 0)
db.head()
```

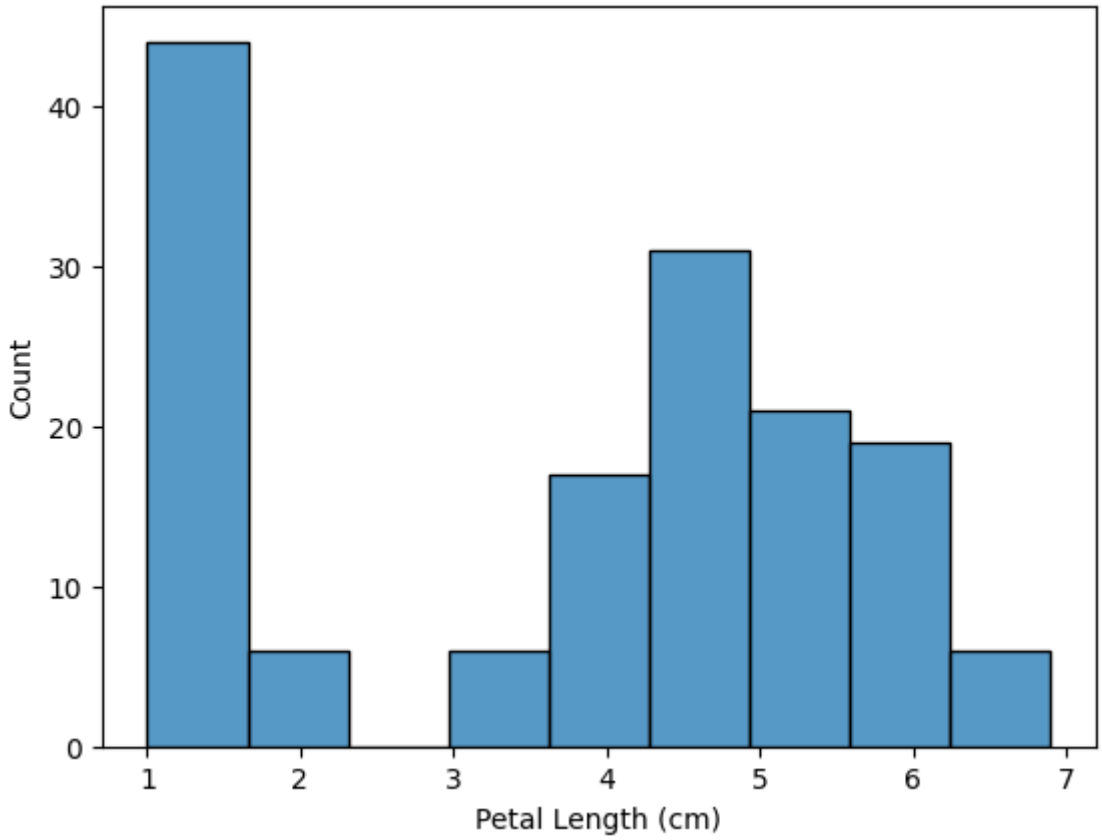
Out[20]:

Id	Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa

Histogram

In [21]:
```
sns.histplot(db["Petal Length (cm)"])
```

Out[21]:
```
<Axes: xlabel='Petal Length (cm)', ylabel='Count'>
```

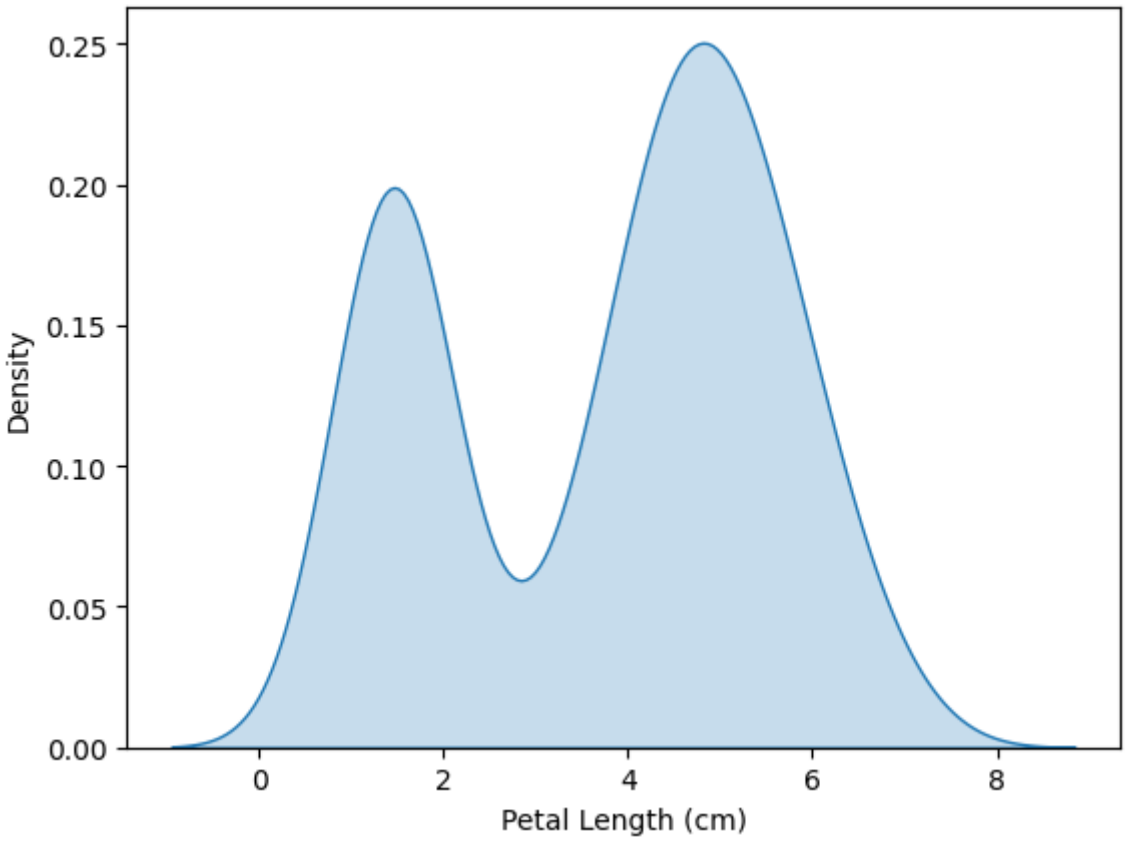


Density

The next type of plot is a **kernel density estimate (KDE)** plot. In case you're not familiar with KDE plots, you can think of it as a smoothed histogram.

In [22]:
```
sns.kdeplot(data = db["Petal Length (cm)"], shade = True)
```

Out[22]:
```
<Axes: xlabel='Petal Length (cm)', ylabel='Density'>
```

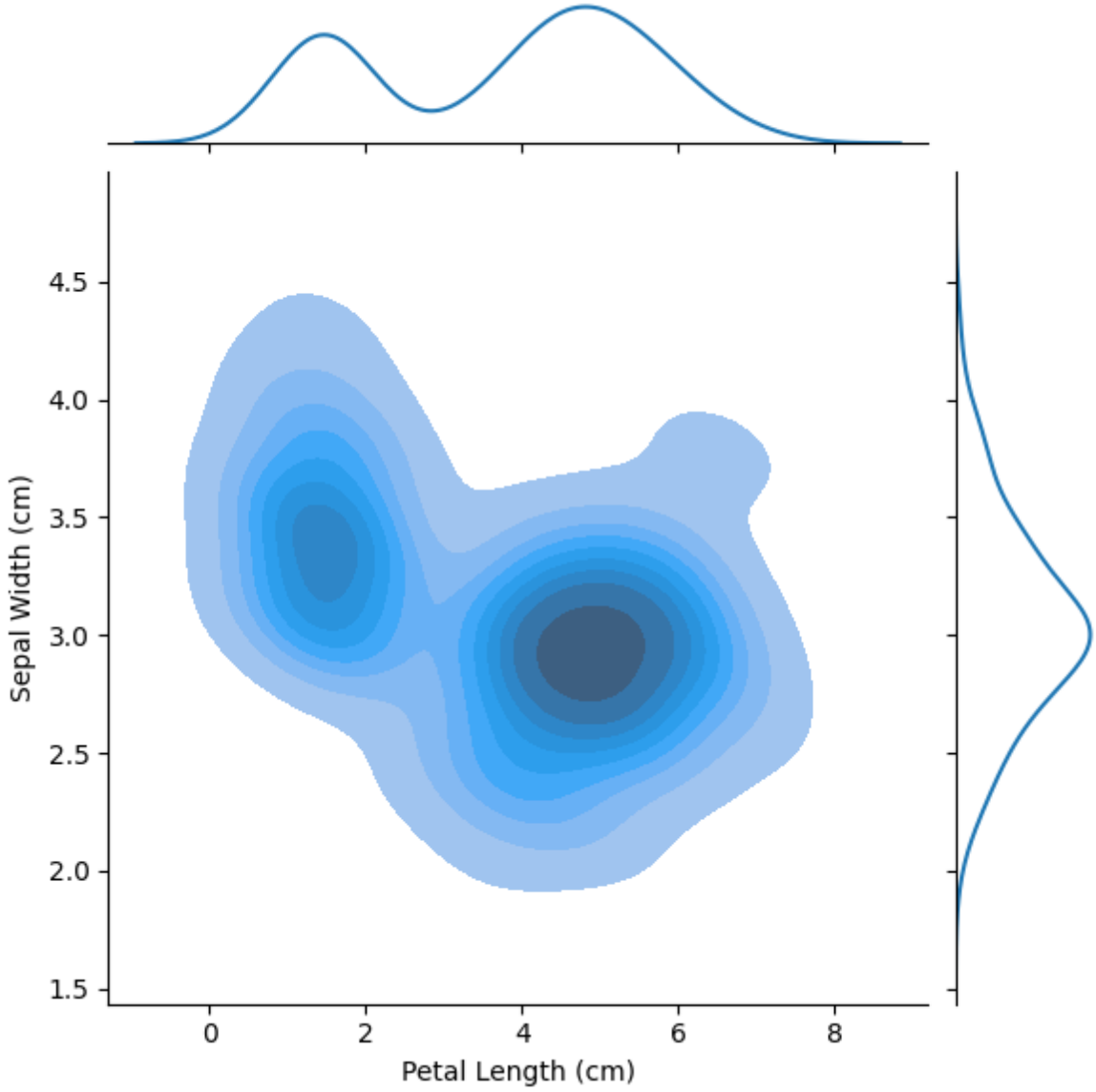


shade is replaced by fill

2D KDE Plots

In [23]:
```
sns.jointplot(x = db["Petal Length (cm)"], y = db["Sepal Width (cm)"], kind = "k
```

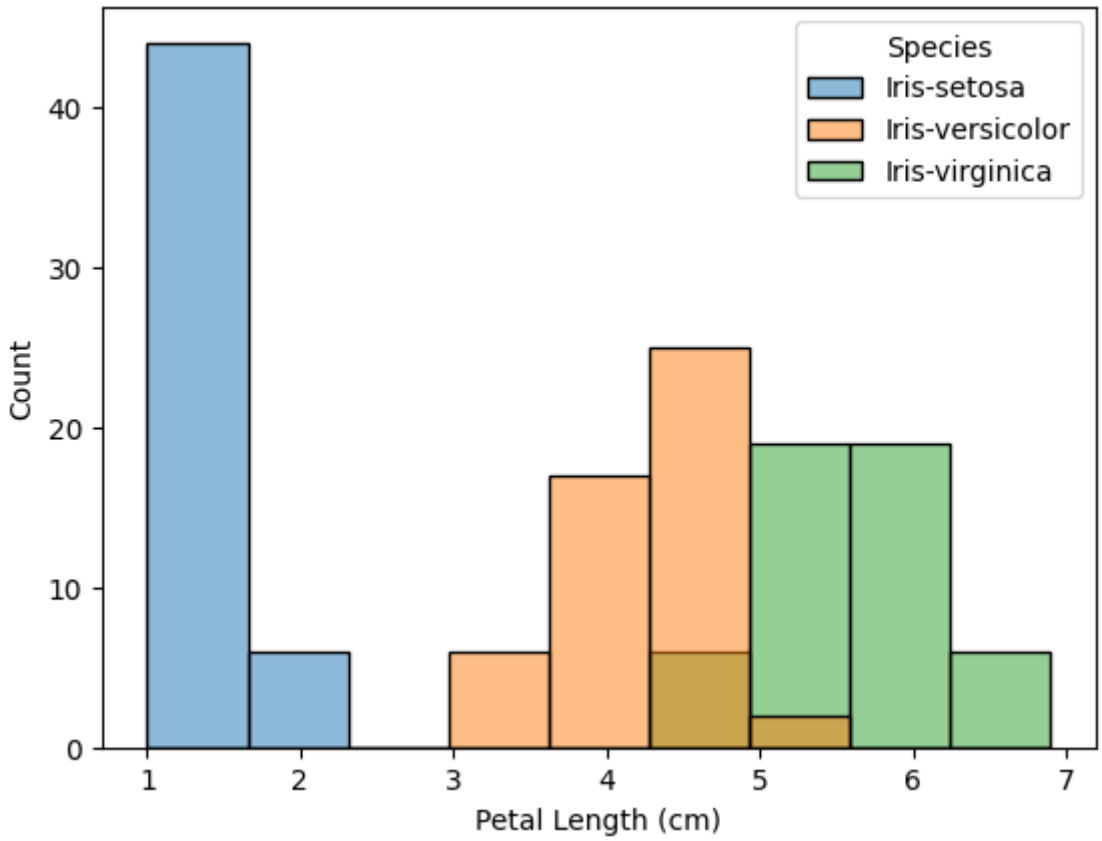
Out[23]:
```
<seaborn.axisgrid.JointGrid at 0x1ff5f7d5f70>
```



Color-Coded Plots

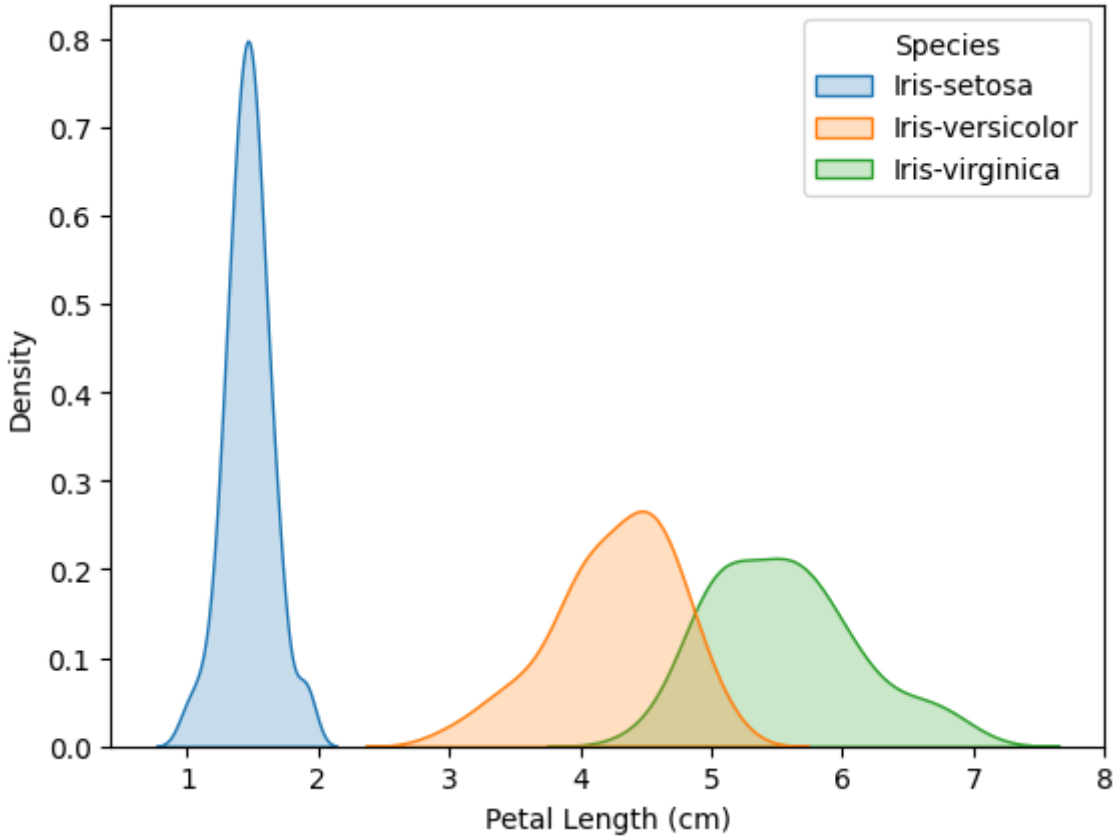
In [24]:
```
sns.histplot(data = db, x = "Petal Length (cm)", hue = "Species")
```

Out[24]:
```
<Axes: xlabel='Petal Length (cm)', ylabel='Count'>
```



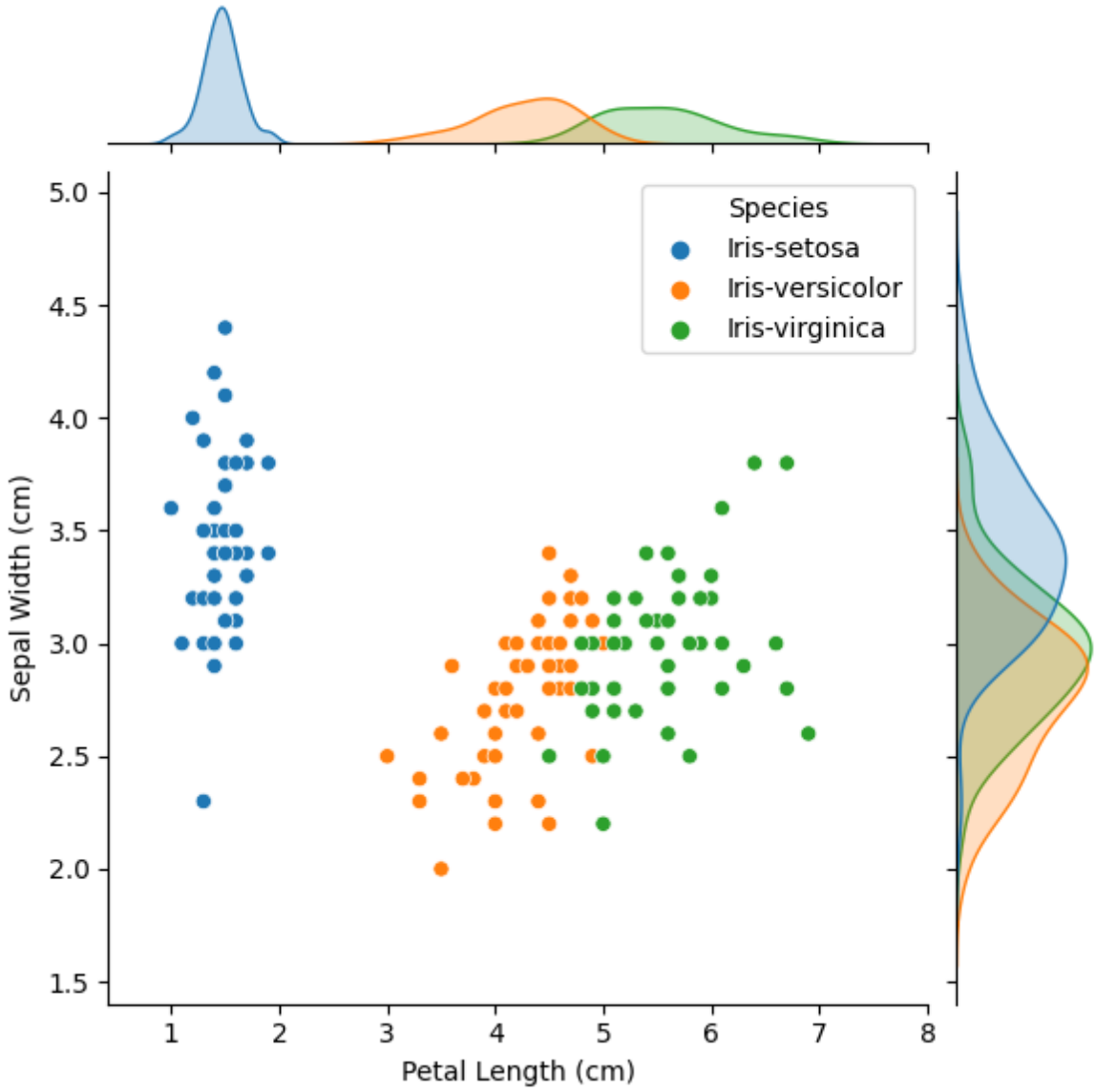
In [27]:
```
sns.kdeplot(data = db, x = "Petal Length (cm)", hue = "Species", fill = True)
```

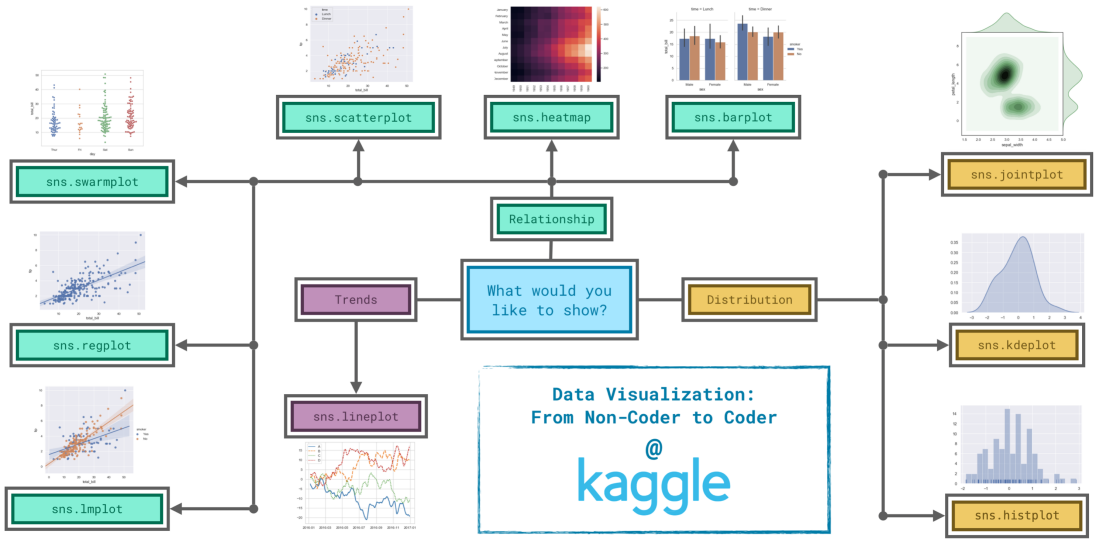
Out[27]:
```
<Axes: xlabel='Petal Length (cm)', ylabel='Density'>
```



In [26]:
```
sns.jointplot(data = db, x = "Petal Length (cm)", y = "Sepal Width (cm)", hue =
```

Out[26]:
```
<seaborn.axisgrid.JointGrid at 0x1ff61ad1d00>
```


Species
Iris-setosa
Iris-versicolor
Iris-virginica
Sepal Width (cm)
Petal Length (cm)
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1
2
3
4
5
6
7
8


sns.scatterplot
sns.heatmap
sns.barplot
sns.swarmplot
sns.jointplot
Relationship
What would you like to show?
Trends
Distribution
sns.regplot
sns.kdeplot
sns.lineplot
sns.lmplot
sns.histplot
Data Visualization:
From Non-Coder to Coder
@
kaggle

In [ ]: