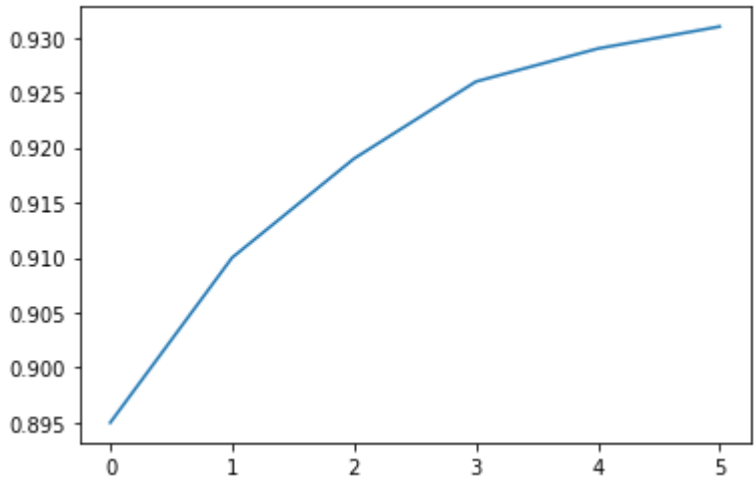
In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [3]:
```
yield_apples = [0.895, 0.91, 0.919, 0.926, 0.929, 0.931]
```

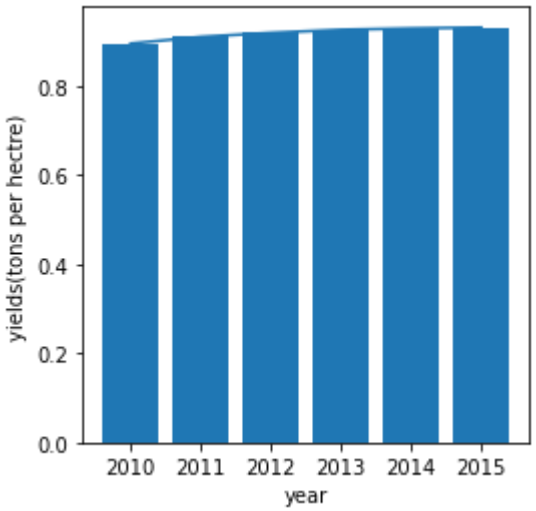
In [ ]:
```

```

In [4]:
```
plt.plot(yield_apples);
```



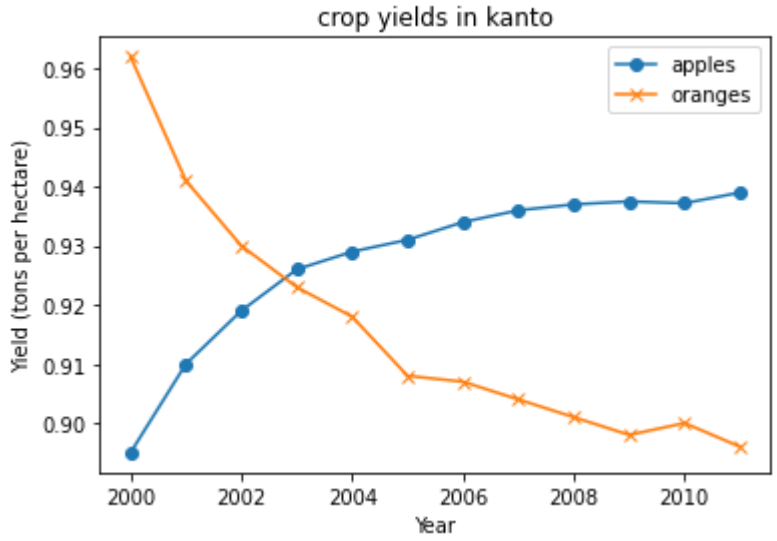
In [5]:
```
years = [2010, 2011, 2012, 2013, 2014, 2015]
yield_apples = [0.895, 0.91, 0.919, 0.926, 0.929, 0.931]
```

In [6]:
```
plt.figure(figsize=(4, 4))
plt.bar(years,yield_apples)
plt.plot(years,yield_apples)
plt.xlabel("year")
plt.ylabel("yields(tons per hectre)");
```

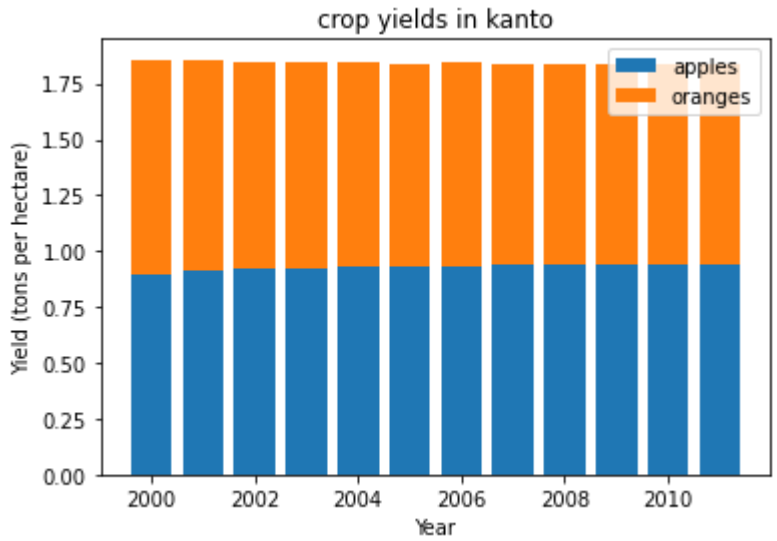


In [7]:
```
years = range(2000, 2012)
apples = [0.895, 0.91, 0.919, 0.926, 0.929, 0.931, 0.934, 0.936, 0.937, 0.9375, 0
oranges = [0.962, 0.941, 0.930, 0.923, 0.918, 0.908, 0.907, 0.904, 0.901, 0.898,
```

In [8]:
```
plt.plot(years, apples, marker="o")
plt.plot(years, oranges, marker="x")
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.legend(["apples", "oranges"])
plt.title("crop yields in kanto");
```

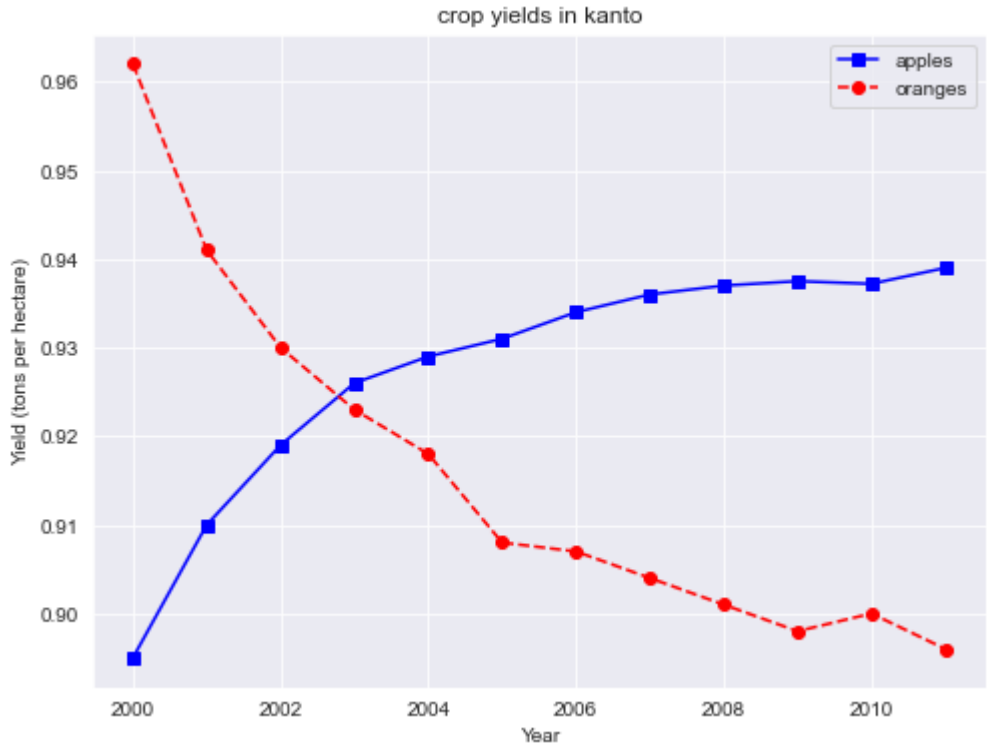


In [9]:
```
plt.bar(years, apples,)
plt.bar(years, oranges, bottom=apples)
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.legend(["apples", "oranges"])
plt.title("crop yields in kanto");
```

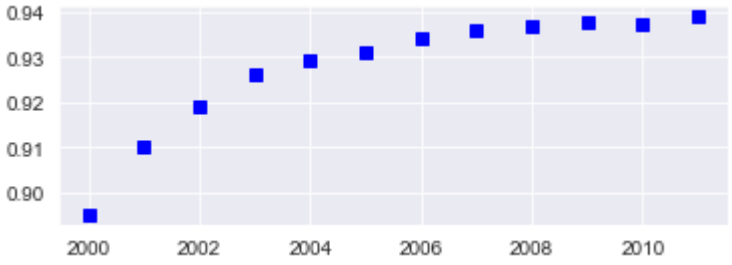


In [10]:
```
sns.set_style("darkgrid")
```

In [11]:
```
plt.figure(figsize=(8, 6))
plt.plot(years, apples, "s-b")
plt.plot(years, oranges, "o--r")
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.legend(["apples", "oranges"])
plt.title("crop yields in kanto");
```



In [12]:
```
plt.figure(figsize=(6, 2))
plt.plot(years, apples, "sb");
```



In [13]:
```
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9, 5)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_5260\1103922644.py in <module>
----> 1 matplotlib.rcParams['font.size'] = 14
      2 matplotlib.rcParams['figure.figsize'] = (9, 5)
      3 matplotlib.rcParams['figure.facecolor'] = '#00000000'

NameError: name 'matplotlib' is not defined
```

In [ ]:
```
matplotlib.rcParams
```

line plot represents bunch of values in a sequence

scatter plot visualizes the relationship between two variables as points on a 2 dimensional grid

In [20]:
```
flowers_df = sns.load_dataset("iris")
```

In [21]:
```
flowers_df
```

Out[21]:

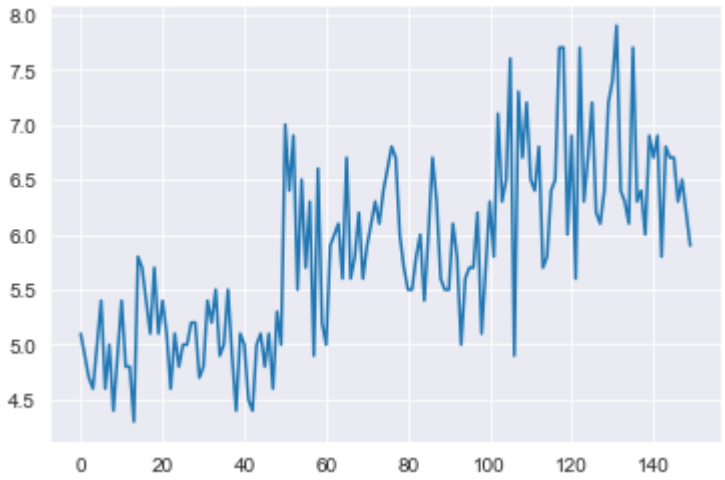
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [22]:
```
flowers_df.species.unique()
```

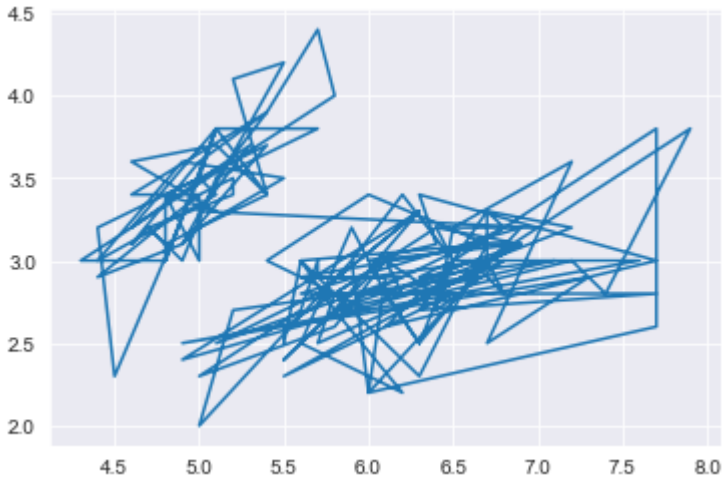
Out[22]:
```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

In [23]:
```
plt.plot(flowers_df.sepal_length);
```

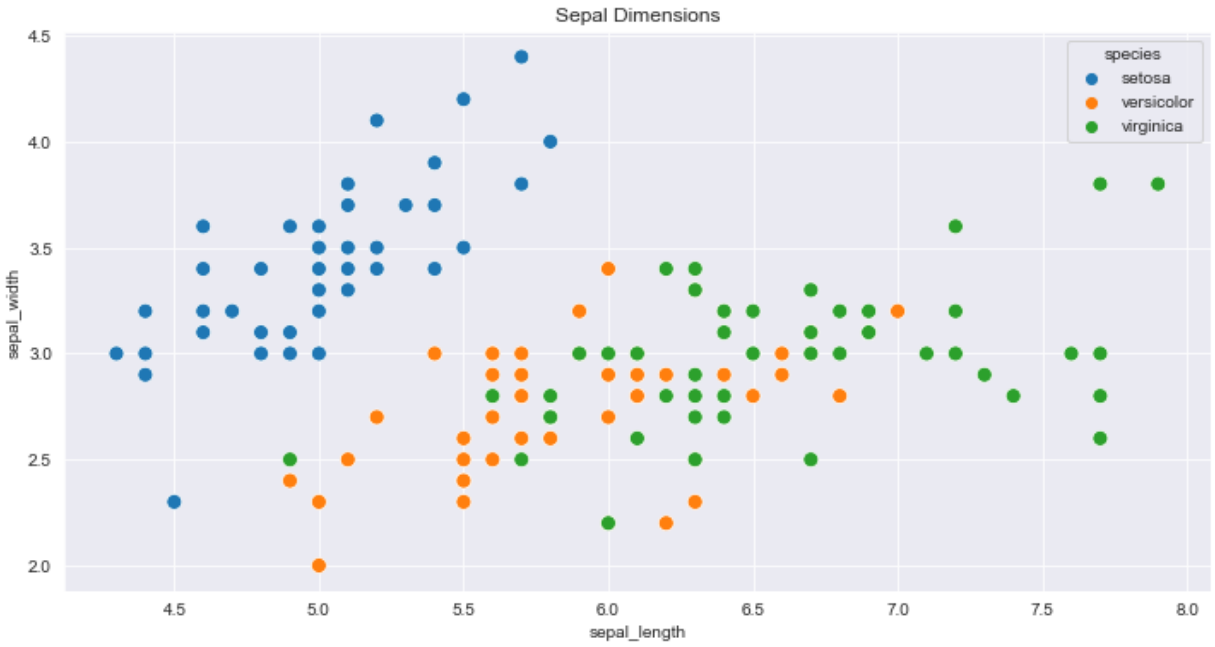


In [24]:
```
plt.plot(flowers_df.sepal_width)
plt.plot(flowers_df.sepal_length);
```

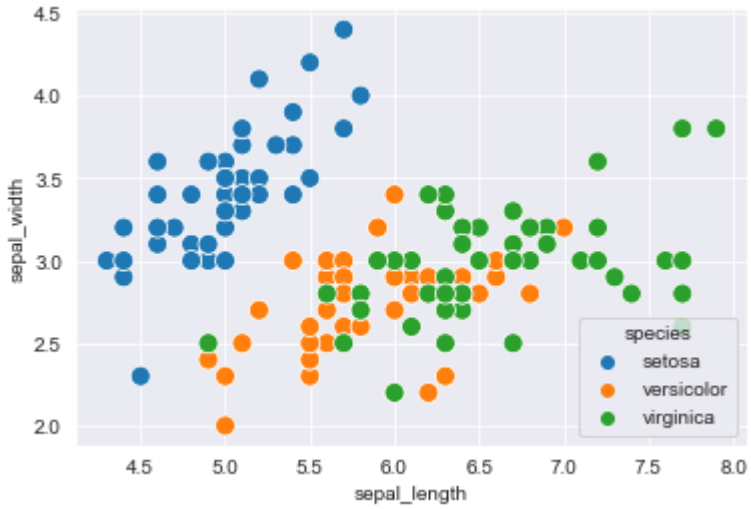
In [25]:
```
plt.plot(flowers_df.sepal_length, flowers_df.sepal_width); #Line graph
```



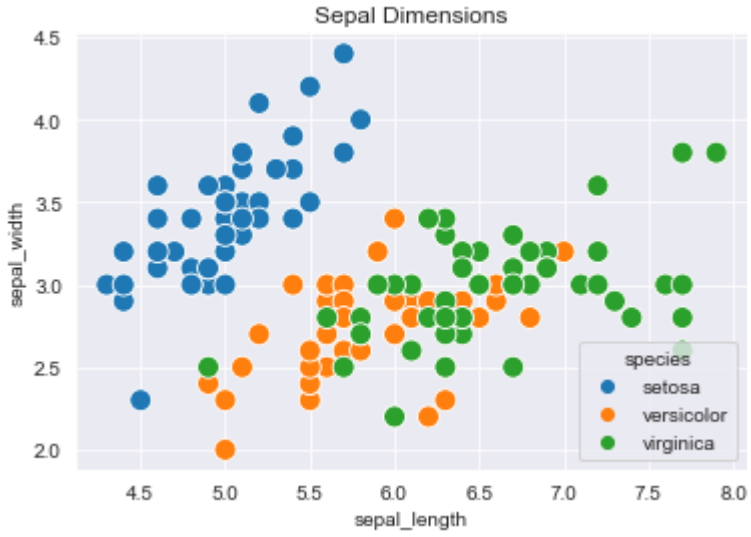
In [26]:
```
plt.figure(figsize=(12, 6))
plt.title('Sepal Dimensions')

sns.scatterplot(x=flowers_df.sepal_length,
                y=flowers_df.sepal_width,
                hue=flowers_df.species,
                s=80);
```



In [27]:
```
sns.scatterplot(x=flowers_df.sepal_length,
                y=flowers_df.sepal_width,
                hue=flowers_df.species,
                s=100);
```



In [28]:
```
plt.title('Sepal Dimensions')
sns.scatterplot(x='sepal_length',
                y='sepal_width',
                hue='species',
                s=120,
                data=flowers_df);
```



In [29]:
```
flowers_df
```

Out[29]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

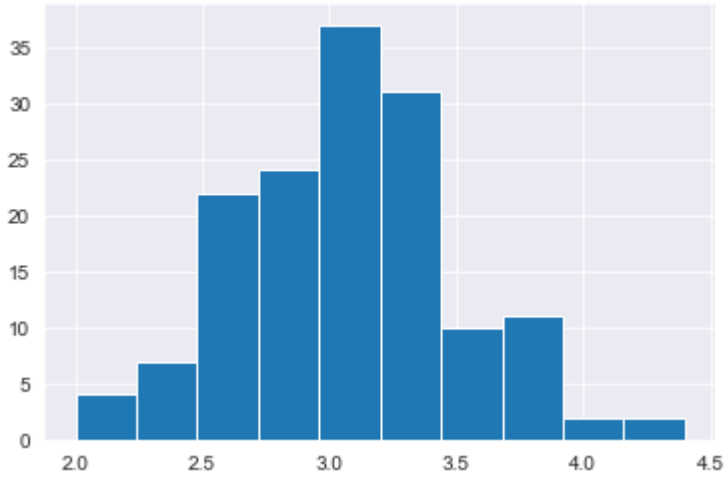
In [30]:
```
flowers_df.species.unique()
```

Out[30]:
```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

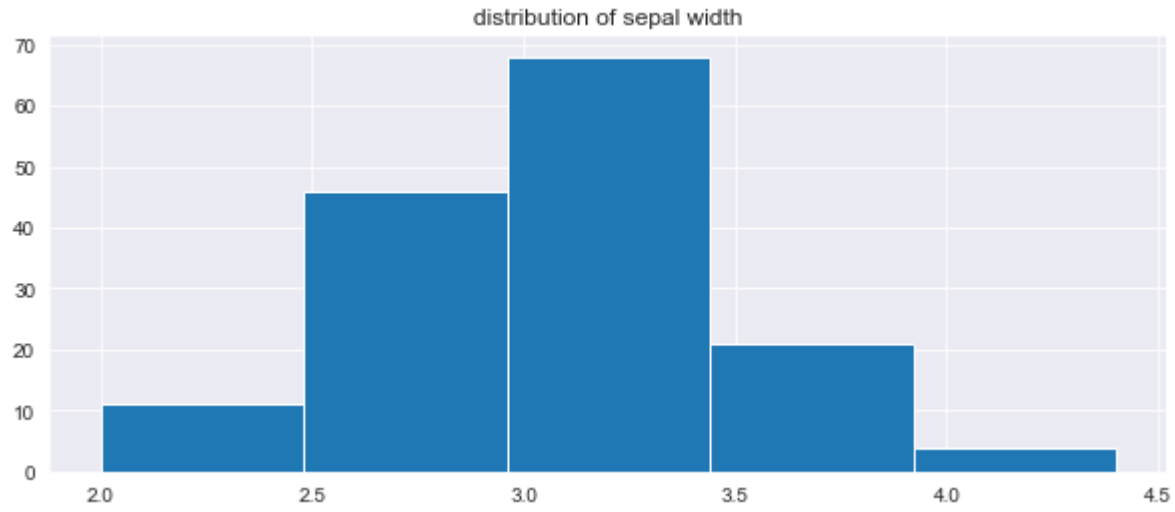
In [31]:
```
flowers_df.sepal_width.mean()
```

Out[31]:
```
3.0573333333333337
```

In [32]:
```
plt.hist(flowers_df.sepal_width);
```



In [33]:
```
plt.figure(figsize=(10,4))
plt.title("distribution of sepal width")
plt.hist(flowers_df.sepal_width, bins=5);
```



In [34]:
```
import numpy as np
```

In [ ]:
```
plt.figure(figsize=(10,4))
plt.title("distribution of sepal width")
plt.hist(flowers_df.sepal_width, bins=np.arange(2, 5, 0.25));
```

In [ ]:
```
setosa_df = flowers_df[flowers_df.species == "setosa"]
versicolor_df = flowers_df[flowers_df.species == "versicolor"]
virginica_df = flowers_df[flowers_df.species == "virginica"]
```

In [ ]:
```
setosa_df
```

In [ ]:
```
plt.title("Distribution of setosa and versicolor")
plt.hist(setosa_df.sepal_width, alpha=0.9, bins=np.arange(2, 5, 0.25));
plt.hist(versicolor_df.sepal_width, alpha=0.7, bins=np.arange(2, 5, 0.25));
plt.legend(["setosa", "versicolor"]);
```

In [ ]:
```
plt.figure(figsize=(10,8))
plt.hist([setosa_df.sepal_width, versicolor_df.sepal_width, virginica_df.sepal_wi
         stacked=True);
plt.legend(["setosa", "versicolor", "virginica"]);
plt.figure(figsize=(8,6));
```

In [ ]:
```
tips_df = sns.load_dataset("tips")
```

In [ ]:
```
tips_df
```

In [ ]:
```
tips_time_df = tips_df[tips_df.time == "Dinner"]
```

In [ ]:
```
tips_time_df
```

In [ ]:
```
tips_time_df.drop[tips_time_df.size]
```

In [ ]:
```
tips_df.time.unique()
```

In [ ]:
```
tips_df
```

In [ ]:
```
bill_avg_df = tips_df.groupby("day")[["total_bill"]].mean()
bill_avg_df
```

In [ ]:
```
bill_avg_df
```

In [ ]:
```
plt.bar(bill_avg_df.index, bill_avg_df.total_bill);
```

In [ ]:
```
sex_avg_df = tips_df.groupby("sex")[["smoker"]].count()
```

In [ ]:
```
sex_avg_df
```

In [ ]:
```
plt.bar(sex_avg_df.index, sex_avg_df.smoker);
```

In [ ]:
```
sns.barplot(x="day", y="total_bill", data=tips_df);
```

In [ ]:
```
flight_schedule = sns.load_dataset("flights")
```

In [ ]:
```
flight_schedule
```

In [ ]:
```
flight_schedule.passengers.count()
```

In [ ]:
```
plt.plot(flight_schedule.passengers)
```

In [ ]:
```
flight_schedules = sns.load_dataset("flights").pivot("month", "year", "passengers
```

In [ ]:
```
flight_schedules
```

In [ ]:
```
plt.title("flight schedule in(1000's)")
sns.heatmap(flight_schedules);
```

In [ ]:
```
plt.title("No. of Passengers (1000s)")
sns.heatmap(flight_schedules, fmt="d", annot=True, cmap='Blues');
```

In [ ]:
```
from urllib.request import urlretrieve
```

In [ ]:
```
urlretrieve('https://i.imgur.com/SkPbq.jpg', './glo/chart.jpg');
```

In [ ]:
```
from PIL import Image
```

In [ ]:
```
img = Image.open('./glo/chart.jpg')
```

In [ ]:
```
plt.imshow(img);
```

In [ ]:
```
plt.grid(False)
plt.title('A data science meme')
plt.axis('off')
plt.imshow(img);
```

In [ ]:
```
image_array = np.array(img)
```

In [ ]:
```
image_array
```

In [ ]:
```
image_array.shape
```

In [ ]:
```
plt.grid(False)
plt.axis('off')
plt.imshow(image_array[125:325,105:305]);
```

In [ ]:
```
fig, axes = plt.subplots(2, 4)

plt.tight_layout(pad=2)
```

In [ ]:
```
axes
```

In [ ]:
```
fig
```

In [ ]:
```
fig, axes = plt.subplots(2, 3, figsize=(16, 8))

# Use the axes for plotting
axes[0,0].plot(years, apples, 's-b')
axes[0,0].plot(years, oranges, 'o--r')
axes[0,0].set_xlabel('Year')
axes[0,0].set_ylabel('Yield (tons per hectare)')
axes[0,0].legend(['Apples', 'Oranges']);
axes[0,0].set_title('Crop Yields in Kanto')


# Pass the axes into seaborn
axes[0,1].set_title('Sepal Length vs. Sepal Width')
sns.scatterplot(x=flowers_df.sepal_length,
                y=flowers_df.sepal_width,
                hue=flowers_df.species,
                s=100,
                ax=axes[0,1]);

# Use the axes for plotting
axes[0,2].set_title('Distribution of Sepal Width')
axes[0,2].hist([setosa_df.sepal_width, versicolor_df.sepal_width, virginica_df.se
         bins=np.arange(2, 5, 0.25),
         stacked=True);

axes[0,2].legend(['Setosa', 'Versicolor', 'Virginica']);

# Pass the axes into seaborn
axes[1,0].set_title('Restaurant bills')
sns.barplot(x='day', y='total_bill', hue='sex', data=tips_df, ax=axes[1,0]);

# Pass the axes into seaborn
axes[1,1].set_title('Flight traffic')
sns.heatmap(flight_schedule, cmap='Blues', ax=axes[1,1]);

# Plot an image using the axes
axes[1,2].set_title('Data Science Meme')
axes[1,2].imshow(img)
axes[1,2].grid(False)
axes[1,2].set_xticks([])
axes[1,2].set_yticks([])

plt.tight_layout(pad=2);
```

In [1]:
```
import pandas as pd
red_wine = pd.read_csv('red_wine.csv', sep = ';', encoding="UTF-8")
red_wine.head()
```

Out[1]:

fixed acidity;"volatile acidity";"ci sugar";"chlorides";"free sulfur di dioxide";"density";"pH";"sulphates";									
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4
7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8
	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4

In [ ]:
```
```