INTRO TO MATPLOTLIB

LECTURE # 1

MATPLOTLIB

library for creating a wide range of plots, including line plots, bar charts, scatter plots, and histograms.

from matplotlib import pyplot as plt

MATPLOTLIB

For displaying the Data

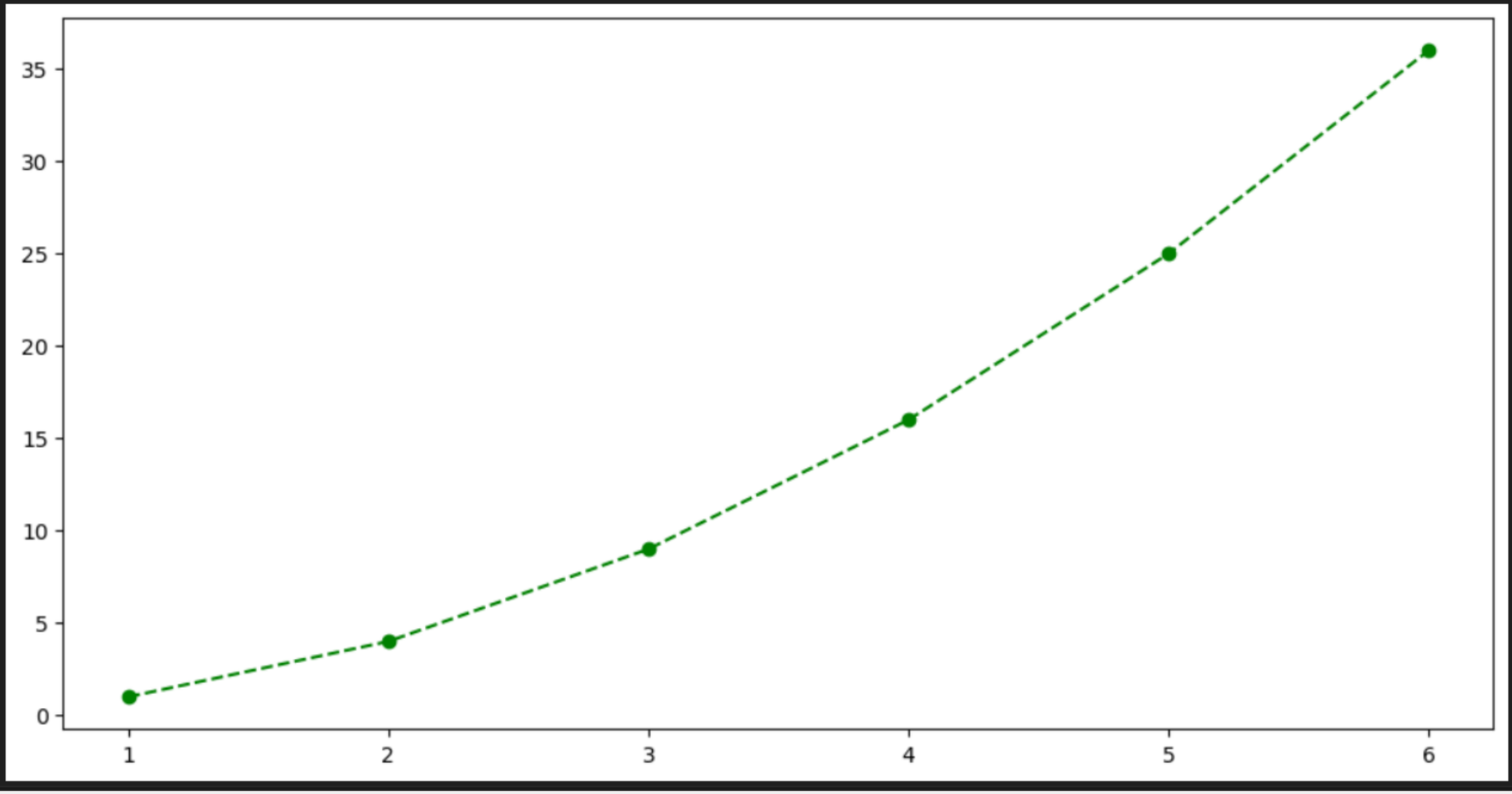
```
plt.figure(figsize=(12,6))
plt.plot()
plt.show()
```

MATPLOTLIB

```
# PLot some data
#Plot the squares against the numbers
plt.figure(figsize=(12,6))
x=[1,2,3,4,5,6]
y=[1,4,9,16,25,36]
plt.plot(x,y)
plt.show()
```

plot(<col name>, <col name>, color= <color name>, marker= <marker>, ls= <line style>)

```
plt.plot(x,y,marker='o',ls='--',color='g')
plt.show()
```

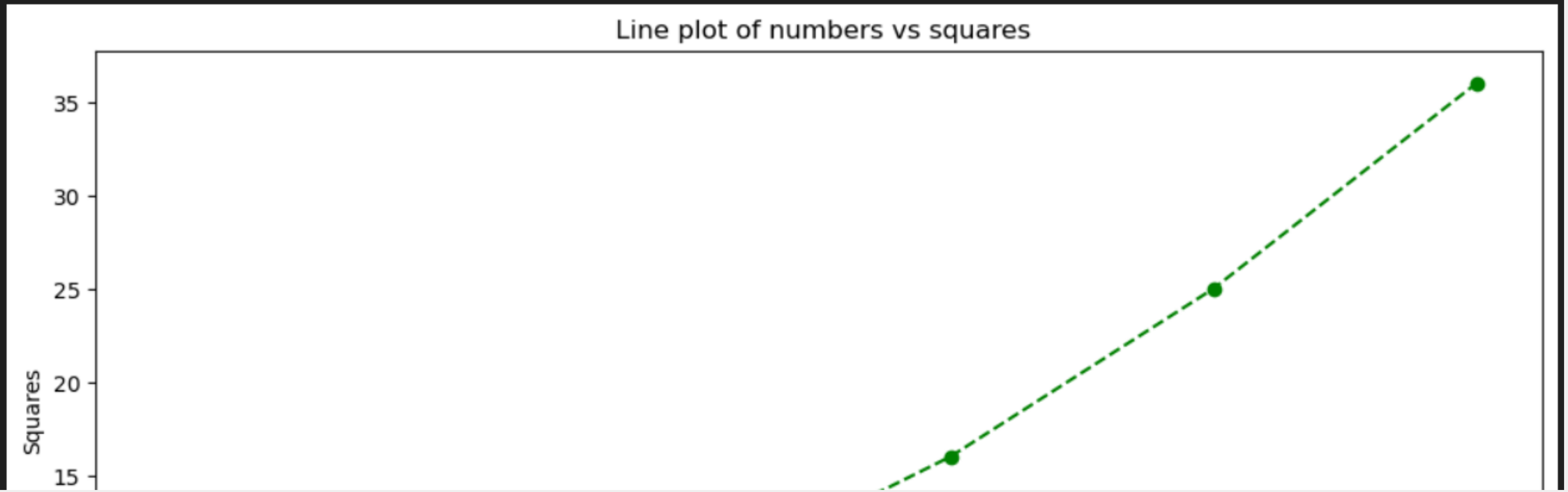


MARKERS

- ".": point
- "o": circle
- "s": square
- "^": triangle
- "v": upside down triangle
- "+": plus
- "x": X

SETTERS

```
plt.plot(x,y,marker='o',ls='--',color='g')
plt.title(" Line plot of numbers vs squares")
plt.xlabel("Numbers")
plt.ylabel("Squares")
plt.show()
```

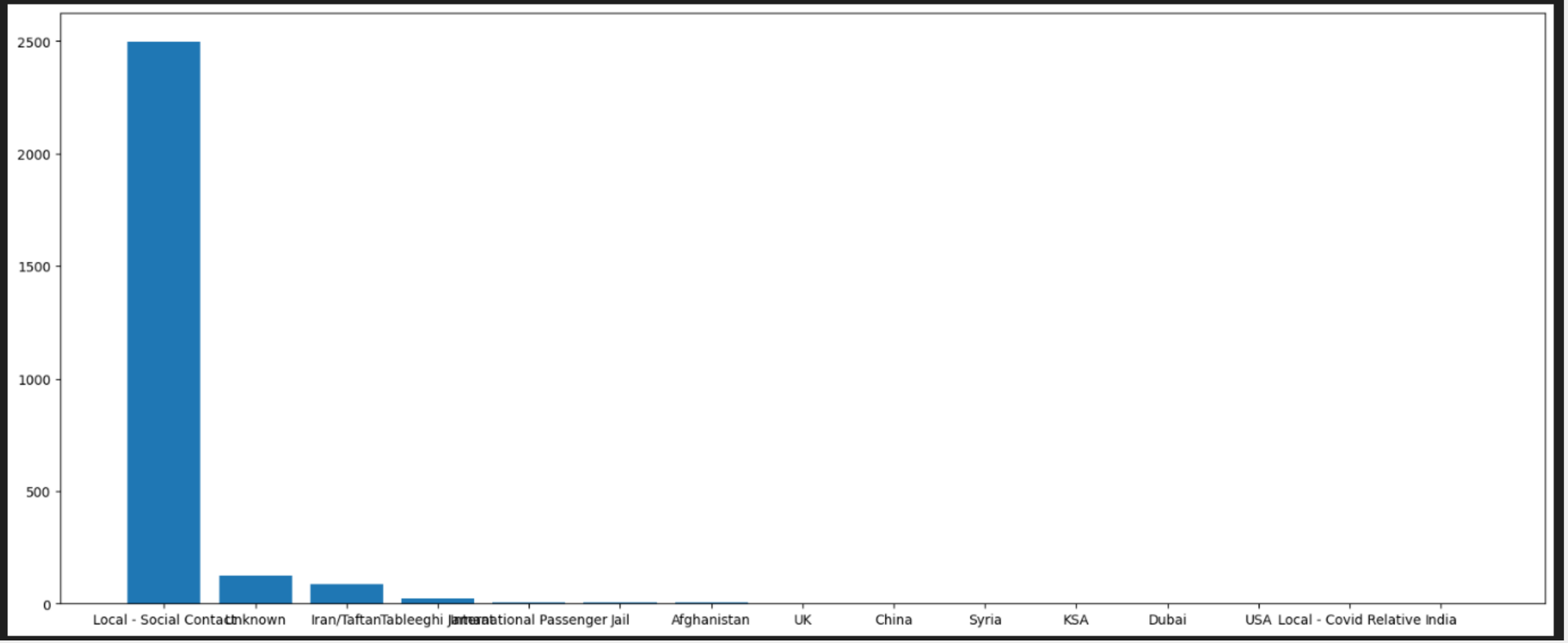


CATEGORICAL DATA

FOR CATEGORICAL COLUMNS WE CAN VIEW THE DISTRIBUTION WITH THE HELP OF BAR PLOT

```
plt.figure(figsize=(20,8))
plt.bar(data['Travel_history'].value_counts().index,data['Travel_history'].value_counts())
plt.show()
```

✓ 0.5s Pyth



CATEGORICAL DATA

FOR CATEGORICAL COLUMNS WE CAN VIEW THE DISTRIBUTION WITH THE HELP OF BAR PLOT

```
plt.figure(figsize=(20,8))
plt.bar(data['Travel_history'].value_counts().index,data['Travel_history'].value_counts())
plt.xticks(rotation="vertical", size=14)
plt.title(" Distribution Plot of Travel History", size=20 ,color='blue')
plt.xlabel("Unique Categories", size=12, color='blue')
plt.ylabel("Count against each category" , size=12, color='blue')
plt.yticks(size=14)
plt.show()
```

NUMERIC DATA

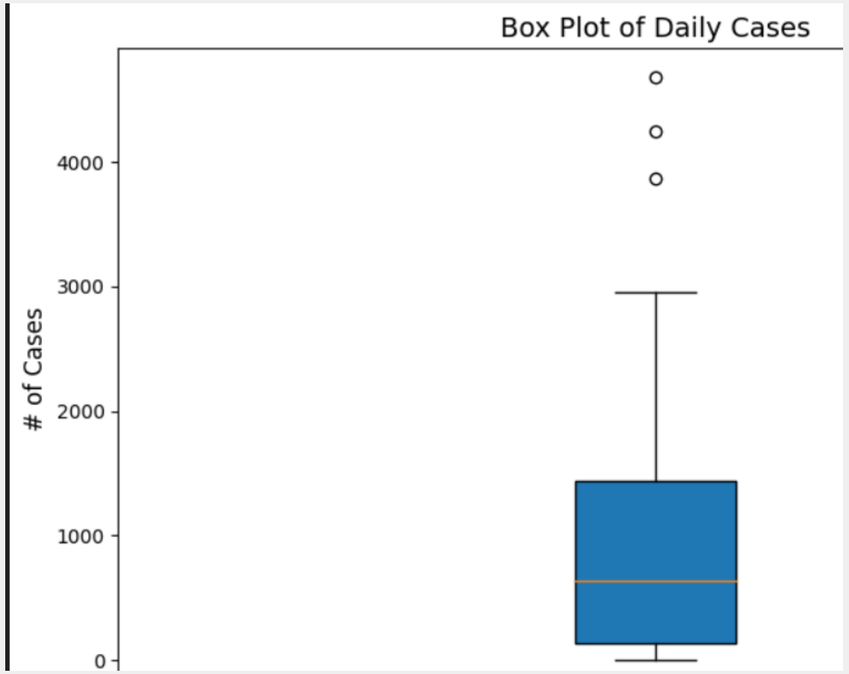
THE DISTRIBUTION OF A NUMERICAL COLUMN CAN BE BEST VIEWED WITH THE HELP OF A BOXPLOT.

```
import numpy as np
daily_data = data.pivot_table(index=['Date'],aggfunc=np.sum)
```

✓ 0.0s

```
plt.figure(figsize=(10,6))
plt.boxplot(daily_data['Cases'],vert=True,labels=['Daily Covid Cases'],patch_artist=True
plt.title("Box Plot of Daily Cases", size=14)
plt.ylabel("# of Cases", size=12)
plt.show()
```

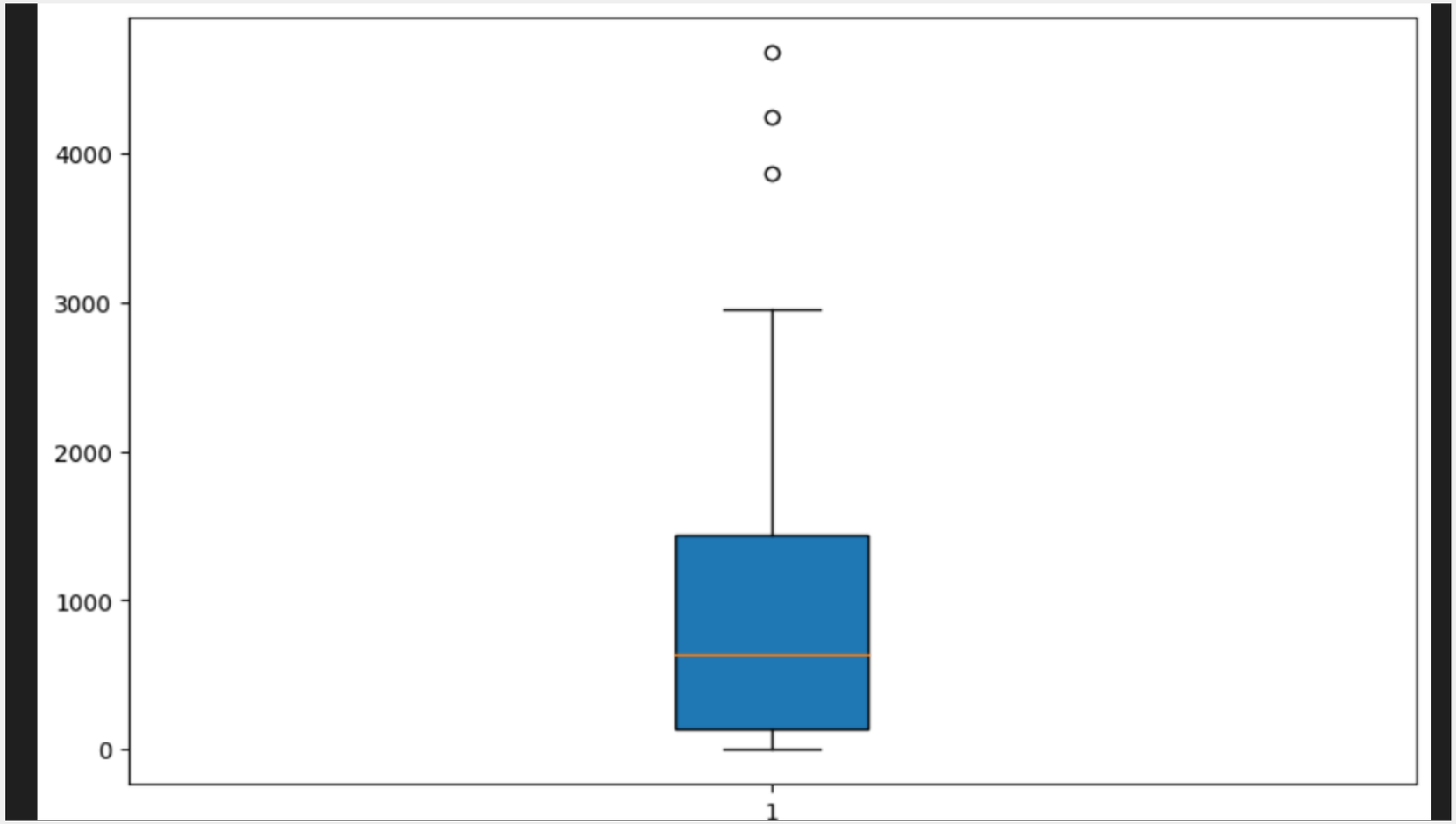
✓ 0.3s



CAN ALSO BE DONE USING GROUPBY

```
plt.figure(figsize=(10,6), )
a = data.groupby("Date")["Cases"].sum()
plt.boxplot(x=a,patch_artist=True)
plt.show()
```

✓ 0.2s



for bivariate analysis refer to the notebook