

```
In [11]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import xlrd
```

```
In [12]: #read data from file

ChurnModeling = pd.read_excel('Data/churn-Modelling.xlsx')

ChurnModeling.head()
```

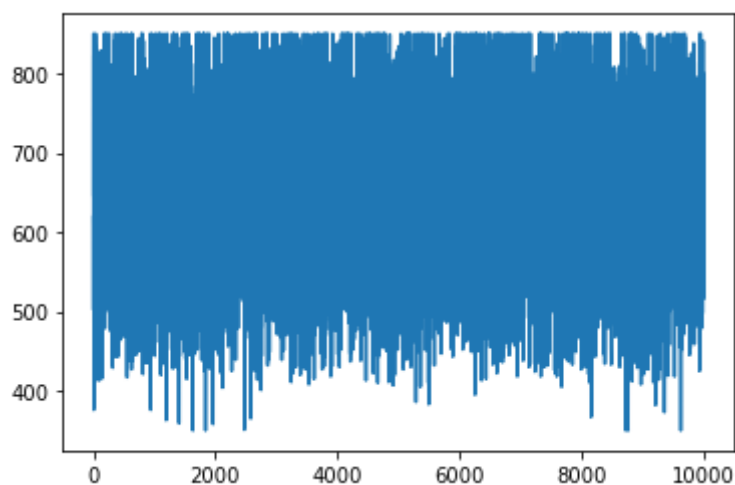
```
Out[12]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenu
<b>0</b>	1	15634602	Hargrave	619	France	Female	42	
<b>1</b>	2	15647311	Hill	608	Spain	Female	41	
<b>2</b>	3	15619304	Onio	502	France	Female	42	
<b>3</b>	4	15701354	Boni	699	France	Female	39	
<b>4</b>	5	15737888	Mitchell	850	Spain	Female	43	

## Scatter Plots of credit scores

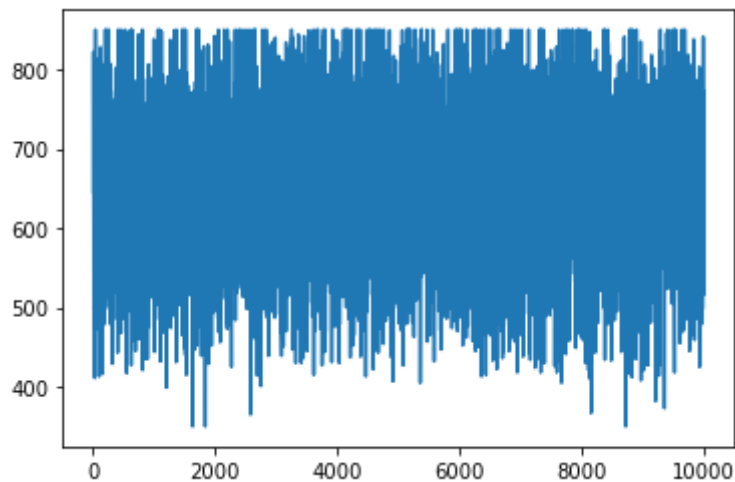
```
In [13]: # Scatter plot of vredit scores
plt.plot(ChurnModeling.CreditScore)
```

```
Out[13]: [<matplotlib.lines.Line2D at 0x11ca6250>]
```



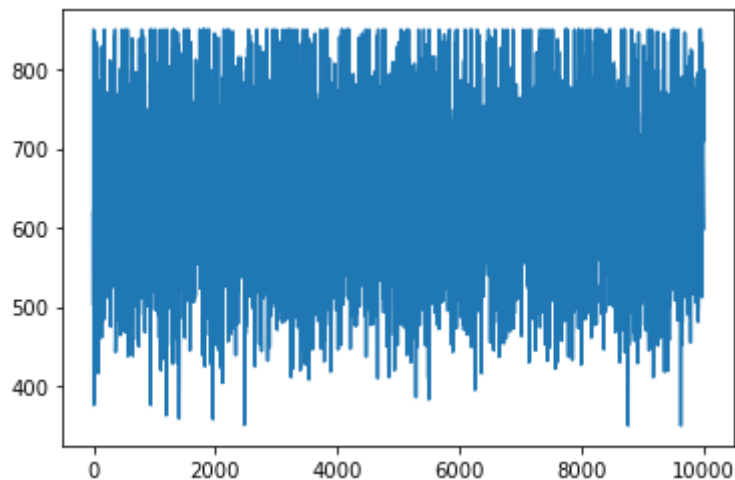
```
In [14]: plt.plot(ChurnModeling[(ChurnModeling["Gender"] == "Male")].CreditScore)
```

```
Out[14]: [<matplotlib.lines.Line2D at 0x11cb2bf0>]
```



```
In [15]: plt.plot(ChurnModeling[(ChurnModeling["Gender"] == "Female")].CreditScore)
```

```
Out[15]: [<matplotlib.lines.Line2D at 0x10a2f5d0>]
```



Analyse churning by evaluating the Exited feature. Start by evaluating Gender feature against Exited Feature with bar charts.

```
In [16]: male_exited = ChurnModeling[(ChurnModeling["Gender"] == "Male") & (ChurnModeling["Exited"] == 1)]  
len(male_exited)
```

```
Out[16]: 898
```

```
In [17]: female_exited = ChurnModeling[(ChurnModeling["Gender"] == "Female") & (ChurnModeling["Exited"] == 1)]  
len(female_exited)
```

```
Out[17]: 1139
```

```
In [18]: width = 0.35      # the width of the bars: can also be len(x) sequence
labels = ["Male", "Female"]
exited = [len(male_exited), len(female_exited)]
stayed = [(len(ChurnModeling["Exited"]) - len(male_exited)) , (len(ChurnModeling["Exited"]) - len(female_exited))]

print(exited)

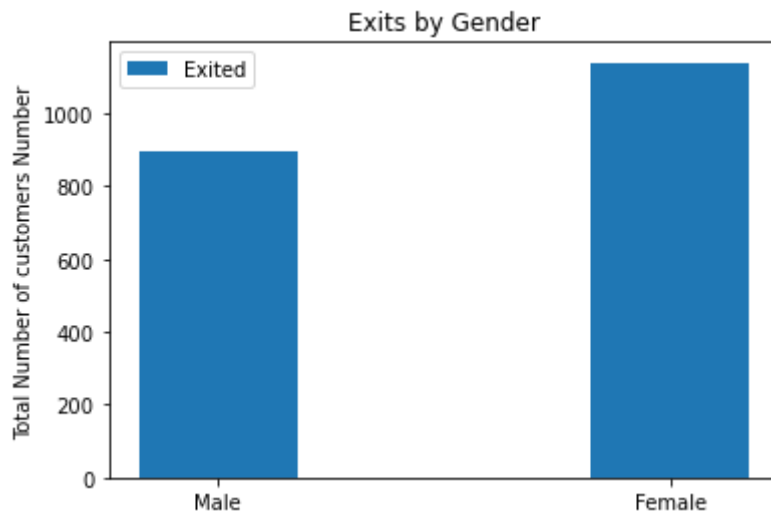
[898, 1139]
```

Bar chart showing the number customers by gender who left the bank

```
In [19]: fig, ax = plt.subplots()

ax.bar(labels, exited, width, label='Exited')
ax.set_ylabel('Total Number of customers Number')
ax.set_title('Exits by Gender')
ax.legend()
```

Out[19]: <matplotlib.legend.Legend at 0x11dc4370>



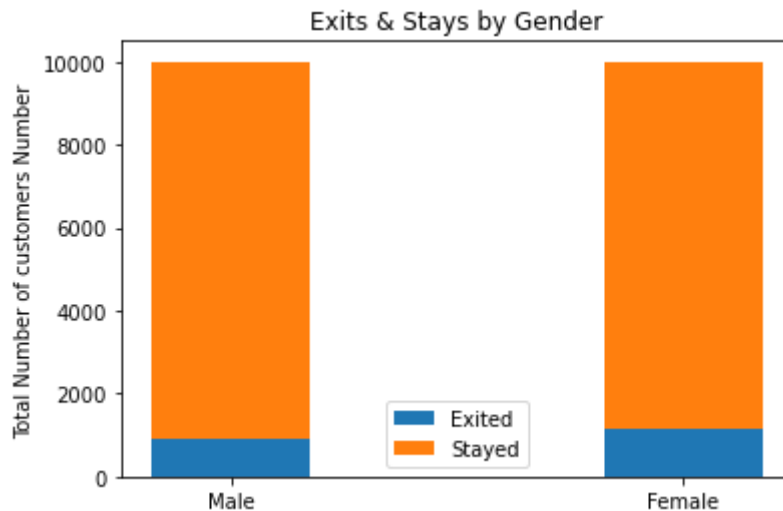
Bar chart showing the number customers by gender who left the bank vs those who stayed. This graph shows both stayed and exited customer based on gender in one glance. However, it is hard to see the difference clearly as the number of those who exited is very close (difference of 241 customers).

```
In [20]: fig, ax = plt.subplots()

ax.bar(labels, exited, width, label='Exited')
ax.bar(labels, stayed, width, bottom=exited, label='Stayed')

ax.set_ylabel('Total Number of customers Number')
ax.set_title('Exits & Stays by Gender')
ax.legend()
```

Out[20]: <matplotlib.legend.Legend at 0x11df8a70>

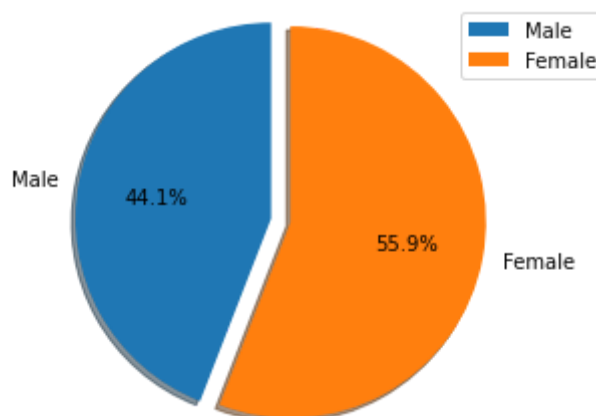


The following pie chart shows the the information as above, but it shows the percenteges and this easier to see the difference.

```
In [21]: fig, ax = plt.subplots()
explode = (0.1, 0.0)
ax.pie(exited, explode=explode, labels=labels, autopct='%1.1f%%',
       shadow=True, startangle=90)
ax.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.

ax.legend()
```

Out[21]: <matplotlib.legend.Legend at 0x1099bfb0>



The following nested pie chart shows two layers: Outer Layer : the total number of customer who exited vs who stayed Inner layer: The customer who exited color coded to distinguish those who stated or exited based on gender. With each gender have a shade of its color to distinguish between genders.

```
In [23]: fig, ax = plt.subplots()
size = 0.4
vals = np.array([exited, stayed])
#print(vals)
cmap = plt.get_cmap("tab20c")
outer_colors = cmap(np.arange(2)*4)
inner_colors = cmap(np.array([1, 2, 5, 6, 9, 10]))
#print("vals.sum", vals.sum(axis=1))
ax.pie(vals.sum(axis=1), radius=1, labels=['Exited', 'Stayed'], colors=outer_colors,
       wedgeprops=dict(width=size, edgecolor='w'))
#print("vals.flatten", vals.flatten())
ax.pie(vals.flatten(), radius=1-size, colors=inner_colors, wedgeprops=dict(
width=size, edgecolor='w'))

ax.set(aspect="equal", title='Nested Pie plot of exits and remains')
```

Out[23]: [Text(0.5, 1.0, 'Nested Pie plot of exits and remains'), None]

