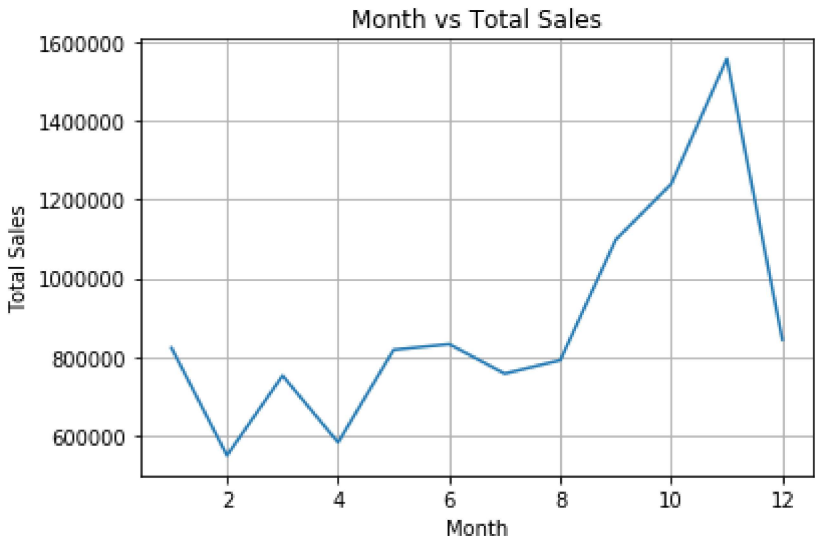
In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt

#query1
df = pd.read_csv("BigMartSalesData.csv")
df_grouped_year = df.query("Year==2011").filter(["Month", "Amount"]).groupby([
"Month"], as_index=False).sum()
df_grouped_year
```

Out[4]:

	Month	Amount
0	1	822669.640
1	2	549134.460
2	3	752003.310
3	4	582318.451
4	5	817655.200
5	6	832231.670
6	7	757108.941
7	8	791173.020
8	9	1097467.722
9	10	1239237.260
10	11	1557236.410
11	12	843909.020

In [5]:
```
#Plot the total sales per month for the year 2011
x = df_grouped_year["Month"]
y = df_grouped_year["Amount"]

plt.plot(x,y)
plt.xlabel("Month")
plt.ylabel("Total Sales")
plt.title("Month vs Total Sales")
plt.grid()
```

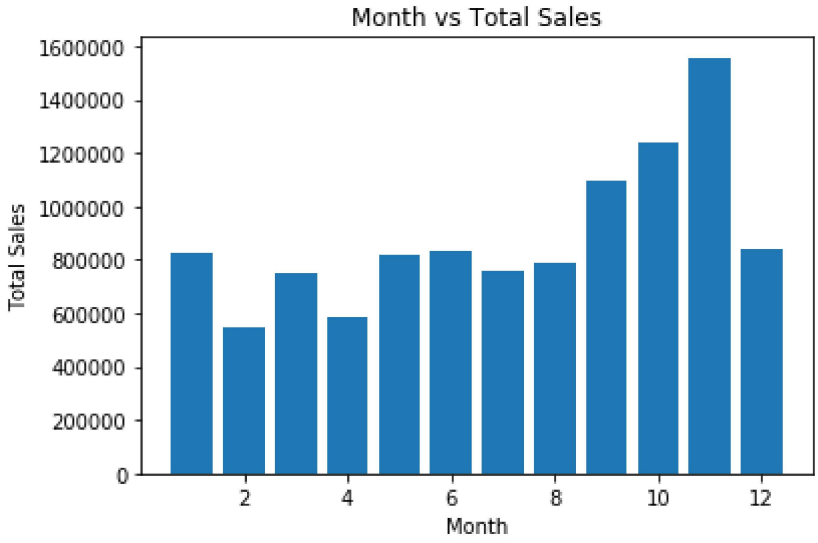


In [6]:
```
#How the total sales have increased over months in Year 2011 ?
#ans- Total Sales have increased till 11th month but got decreased in 12th mon
th
```

In [7]:
```
#Which month has lowest Sales?
#ans- By looking at the graph, we can clearly say that the sales is lowest in
 the 2nd month
```

In [9]:
```
#query2-
plt.bar(x,y)
plt.xlabel("Month")
plt.ylabel("Total Sales")
plt.title("Month vs Total Sales")
```

Out[9]:
```
Text(0.5, 1.0, 'Month vs Total Sales')
```



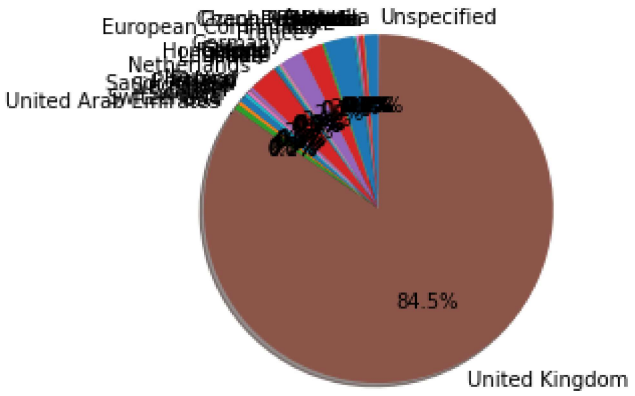
In [10]:
```
#Is Bar Chart Better to visualize than Simple Plot?
# Bar graphs are used to compare things between different groups or to track c
hanges over time.
# However, when trying to measure change over time, bar graphs are best when t
he changes are larger.
```

In [19]:

```
#query3
df_grouped_country = df.query("Year==2011").filter(["Country", "Amount"]).grou
pby(["Country"], as_index=False).sum()

plt.pie(df_grouped_country["Amount"], labels= df_grouped_country["Country"], a
utopct='%1.1f%%', shadow=True, startangle=90)
```

```
Out[19]: ([<matplotlib.patches.Wedge at 0x234b56c9dc8>,
           <matplotlib.patches.Wedge at 0x234b56d0948>,
           <matplotlib.patches.Wedge at 0x234b56d8588>,
           <matplotlib.patches.Wedge at 0x234b56e2248>,
           <matplotlib.patches.Wedge at 0x234b56e8788>,
           <matplotlib.patches.Wedge at 0x234b56eec08>,
           <matplotlib.patches.Wedge at 0x234b56f5f08>,
           <matplotlib.patches.Wedge at 0x234b5701388>,
           <matplotlib.patches.Wedge at 0x234b56e8608>,
           <matplotlib.patches.Wedge at 0x234b56eea88>,
           <matplotlib.patches.Wedge at 0x234b554f588>,
           <matplotlib.patches.Wedge at 0x234b5721388>,
           <matplotlib.patches.Wedge at 0x234b57267c8>,
           <matplotlib.patches.Wedge at 0x234b572dc08>,
           <matplotlib.patches.Wedge at 0x234b573b088>,
           <matplotlib.patches.Wedge at 0x234b57414c8>,
           <matplotlib.patches.Wedge at 0x234b5746908>,
           <matplotlib.patches.Wedge at 0x234b574cd48>,
           <matplotlib.patches.Wedge at 0x234b57591c8>,
           <matplotlib.patches.Wedge at 0x234b5760608>,
           <matplotlib.patches.Wedge at 0x234b5764a48>,
           <matplotlib.patches.Wedge at 0x234b576be88>,
           <matplotlib.patches.Wedge at 0x234b5777308>,
           <matplotlib.patches.Wedge at 0x234b577e748>,
           <matplotlib.patches.Wedge at 0x234b5784b88>,
           <matplotlib.patches.Wedge at 0x234b578bfc8>,
           <matplotlib.patches.Wedge at 0x234b5798448>,
           <matplotlib.patches.Wedge at 0x234b579c888>,
           <matplotlib.patches.Wedge at 0x234b57a4cc8>,
           <matplotlib.patches.Wedge at 0x234b57b1148>,
           <matplotlib.patches.Wedge at 0x234b57b6588>,
           <matplotlib.patches.Wedge at 0x234b57bc9c8>,
           <matplotlib.patches.Wedge at 0x234b57c3dc8>,
           <matplotlib.patches.Wedge at 0x234b57cf248>,
           <matplotlib.patches.Wedge at 0x234b57d3688>,
           <matplotlib.patches.Wedge at 0x234b57dbac8>,
           <matplotlib.patches.Wedge at 0x234b57e0f08>],
          [Text(-0.045092957193071215, 1.0990753501064356, 'Australia'),
           Text(-0.09332845055475869, 1.0960336675107423, 'Austria'),
           Text(-0.09678993099797349, 1.0957334115821273, 'Bahrain'),
           Text(-0.10985905590911363, 1.0945003370647075, 'Belgium'),
           Text(-0.12303820911551719, 1.0930972505214924, 'Brazil'),
           Text(-0.12459019532396202, 1.0929214442168922, 'Canada'),
           Text(-0.13236806902344825, 1.092006728139989, 'Channel Islands'),
           Text(-0.14303011228015966, 1.090661444711935, 'Cyprus'),
           Text(-0.14740428402798303, 1.0900788857005708, 'Czech Republic'),
           Text(-0.1534539053803066, 1.089243727970711, 'Denmark'),
           Text(-0.25249657032221373, 1.0706285452833395, 'EIRE'),
           Text(-0.3443192604767514, 1.0447220907326231, 'European Community'),
           Text(-0.351460393005792, 1.0423413990378654, 'Finland'),
           Text(-0.422639723722009, 1.0155666713378222, 'France'),
           Text(-0.548506409746497, 0.953488709145005, 'Germany'),
           Text(-0.6105620420540817, 0.9149939851183448, 'Greece'),
           Text(-0.6175875337066602, 0.9102667950717114, 'Hong Kong'),
           Text(-0.6242523345742845, 0.9057091270261969, 'Iceland'),
           Text(-0.6273857377060241, 0.9035414412870437, 'Israel'),
           Text(-0.6341383487973756, 0.8988150836432031, 'Italy'),
```

```
 Text(-0.6470819294211244, 0.8895420038517772, 'Japan'),
 Text(-0.6558421194425709, 0.8831031164960731, 'Lebanon'),
 Text(-0.6570510200839793, 0.8822040336603559, 'Malta'),
 Text(-0.7277404840032555, 0.8248598595777998, 'Netherlands'),
 Text(-0.8002596099658291, 0.7547082593011284, 'Norway'),
 Text(-0.8092447025758467, 0.7450657765277705, 'Poland'),
 Text(-0.8186012022641679, 0.734773483225721, 'Portugal'),
 Text(-0.8264984323904044, 0.7258790128225256, 'RSA'),
 Text(-0.8267475889060601, 0.7255952206554397, 'Saudi Arabia'),
 Text(-0.8339036085046058, 0.7173595832795413, 'Singapore'),
 Text(-0.854708578139475, 0.6924400670489809, 'Spain'),
 Text(-0.8751037832103813, 0.6664783331893679, 'Sweden'),
 Text(-0.8928089449012171, 0.6425668742663099, 'Switzerland'),
 Text(-0.9044044870546336, 0.6261409775724994, 'USA'),
 Text(-0.9057576303085958, 0.6241819567544045, 'United Arab Emirates'),
 Text(0.5104736600364763, -0.9743801323964709, 'United Kingdom'),
 Text(0.0015427415139254714, 1.0999989181579324, 'Unspecified')],
[Text(-0.024596158468947932, 0.5994956455126011, '1.3%'),
 Text(-0.05090642757532292, 0.5978365459149503, '0.1%'),
 Text(-0.052794507817076446, 0.5976727699538875, '0.0%'),
 Text(-0.059923121404971066, 0.5970001838534768, '0.4%'),
 Text(-0.06711175042664573, 0.5962348639208139, '0.0%'),
 Text(-0.06795828835852473, 0.5961389695728502, '0.0%'),
 Text(-0.07220076492188086, 0.595640033530903, '0.2%'),
 Text(-0.07801642488008709, 0.5949062425701462, '0.1%'),
 Text(-0.08040233674253619, 0.5945884831094022, '0.0%'),
 Text(-0.08370213020743995, 0.5941329425294787, '0.2%'),
 Text(-0.13772540199393474, 0.5839792065181851, '2.7%'),
 Text(-0.18781050571459162, 0.5698484131268853, '0.0%'),
 Text(-0.19170566891225016, 0.5685498540206537, '0.2%'),
 Text(-0.23053075839382306, 0.5539454570933574, '2.0%'),
 Text(-0.2991853144071 8016, 0.52008475044273, '2.1%'),
 Text(-0.3330338411204082, 0.4990876282463699, '0.0%'),
 Text(-0.33686592747636007, 0.49650916094820613, '0.2%'),
 Text(-0.3405012734041552, 0.4940231601961073, '0.0%'),
 Text(-0.34221040238510403, 0.49284078615656923, '0.1%'),
 Text(-0.3458936447985685, 0.4902627728962925, '0.2%'),
 Text(-0.3529537796842497, 0.4852047293736966, '0.3%'),
 Text(-0.35773206515049316, 0.481692608997858, '0.0%'),
 Text(-0.3583914655003523, 0.4812022001783759, '0.0%'),
 Text(-0.3969493549108666, 0.44992355976970894, '2.6%'),
 Text(-0.4365052417995431, 0.41165905052788815, '0.3%'),
 Text(-0.44140620140500725, 0.40639951446969297, '0.1%'),
 Text(-0.44650974668954607, 0.4007855363049387, '0.3%'),
 Text(-0.45081732675840236, 0.39593400699410486, '0.0%'),
 Text(-0.45095323031239637, 0.3957792112666034, '0.0%'),
 Text(-0.4548565137297849, 0.3912870454252043, '0.3%'),
 Text(-0.4662046789851681, 0.3776945820267168, '0.6%'),
 Text(-0.47732933629657154, 0.36353363628510976, '0.3%'),
 Text(-0.4869866972188456, 0.3504910223270781, '0.5%'),
 Text(-0.49331153839343644, 0.34153144231227234, '0.1%'),
 Text(-0.4940496165319613, 0.3404628855024024, '0.0%'),
 Text(0.27844017820171435, -0.5314800722162568, '84.5%'),
 Text(0.0008414953712320752, 0.5999994099043267, '0.0%')])
```



In [18]:
```
# Which Country contributes highest towards sales?
# United Kingdom
```

In [20]:
```
#query4-
invoice_amounts = df.filter(["InvoiceDate", "Amount"]).groupby("InvoiceDate",
as_index=False).sum()
invoice_amounts
```

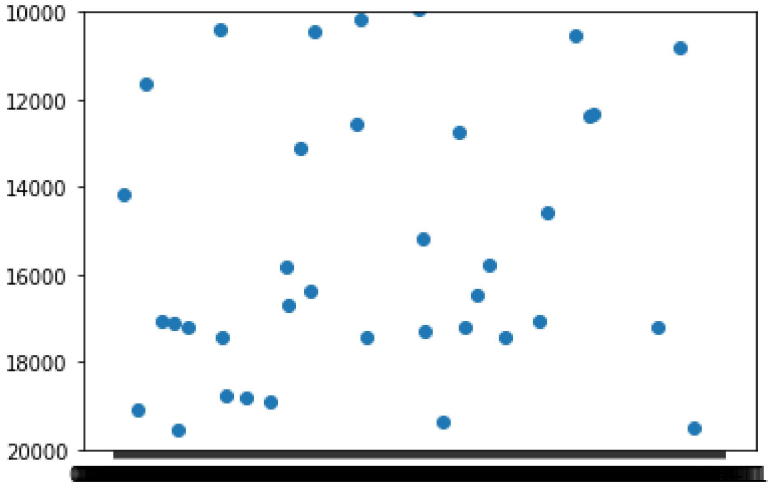
Out[20]:

	InvoiceDate	Amount
0	01-02-11	29636.76
1	01-03-11	27238.21
2	01-04-11	26943.24
3	01-05-11	6982.66
4	01-06-11	21000.18
...	...	...
300	31-03-11	37667.72
301	31-05-11	23435.16
302	31-07-11	33494.86
303	31-08-11	34625.68
304	31-10-11	66246.18

305 rows × 2 columns

In [21]:
```
plt.scatter(invoice_amounts["InvoiceDate"], invoice_amounts["Amount"])
plt.ylim(20000, 10000)
```

Out[21]: (20000, 10000)



In [ ]: