

In [1]:

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from sklearn.cluster import KMeans
```

In [2]:

```
data = pd.read_csv(r'C:/Users/Pranav/Desktop/Prachi/spyderdata/insuranceFraud.csv')
```

In [3]:

```
data.head()
```

Out[3]:

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_lim
0	328	48	521585	10/17/2014	OH	250/500	1000	1406.91	
1	228	42	342868	6/27/2006	IN	250/500	2000	1197.22	500000
2	134	29	687698	09-06-2000	OH	100/300	2000	1413.14	500000
3	256	41	227811	5/25/1990	IL	250/500	2000	1415.74	600000
4	228	44	367455	06-06-2014	IL	500/1000	1000	1583.91	600000

5 rows × 39 columns

In [4]:

```
x = data[['months_as_customer', 'policy_annual_premium', 'policy_deductable']]
```

In [5]:

```
x.head()
```

Out[5]:

	months_as_customer	policy_annual_premium	policy_deductable
0	328	1406.91	1000
1	228	1197.22	2000
2	134	1413.14	2000
3	256	1415.74	2000
4	228	1583.91	1000

In [6]:

```
kmeans = KMeans(7)
kmeans.fit(x)
```

Out[6]:

```
KMeans(n_clusters=7)
```

In [7]:

```
identified_clusters = kmeans.fit_predict(x)
identified_clusters
```

Out[7]:

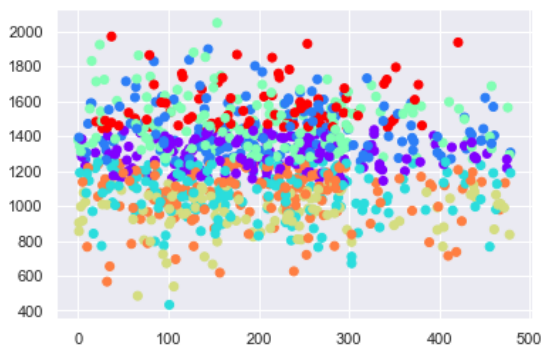
```
array([3, 5, 1, 1, 3, 3, 3, 2, 6, 0, 0, 5, 0, 2, 1, 5, 2, 4, 0, 1, 2, 5,
       5, 1, 2, 1, 1, 4, 1, 5, 2, 1, 2, 3, 0, 3, 2, 5, 5, 0, 6, 6, 6, 4,
       5, 3, 0, 1, 2, 2, 1, 5, 3, 0, 3, 3, 4, 3, 1, 5, 4, 2, 2, 6, 1, 0,
       0, 4, 2, 3, 4, 5, 2, 6, 3, 3, 2, 0, 3, 4, 1, 1, 2, 1, 5, 1, 2, 6,
       1, 2, 1, 5, 1, 3, 6, 5, 3, 1, 2, 2, 6, 3, 6, 3, 6, 0, 2, 5, 1, 2,
       0, 0, 0, 2, 5, 2, 1, 5, 1, 4, 5, 6, 0, 5, 4, 3, 1, 0, 1, 2, 3, 4,
       4, 3, 1, 1, 3, 3, 0, 2, 5, 5, 2, 5, 1, 4, 0, 0, 3, 4, 2, 2, 0, 3,
       2, 3, 1, 0, 2, 0, 2, 5, 3, 2, 6, 3, 1, 3, 6, 5, 4, 5, 3, 0, 2, 5,
       6, 2, 4, 4, 1, 5, 0, 0, 2, 4, 5, 1, 4, 2, 1, 2, 1, 2, 5, 3, 1, 6,
       0, 1, 5, 6, 1, 4, 5, 0, 5, 0, 6, 4, 1, 5, 3, 5, 1, 3, 5, 2, 0, 3,
       1, 6, 2, 0, 5, 2, 2, 6, 3, 2, 2, 4, 5, 1, 1, 2, 5, 1, 3, 3, 1,
       5, 1, 2, 3, 3, 0, 2, 5, 0, 6, 0, 4, 3, 4, 0, 1, 0, 1, 3, 5, 6, 1,
       5, 0, 3, 5, 4, 1, 1, 2, 5, 4, 1, 0, 5, 0, 5, 4, 3, 3, 1, 1, 1, 0,
       2, 0, 2, 2, 0, 5, 1, 3, 4, 0, 5, 4, 0, 5, 4, 5, 1, 0, 0, 4, 0, 0,
       2, 5, 1, 5, 1, 3, 1, 3, 3, 1, 4, 3, 1, 1, 6, 4, 2, 5, 1, 3, 0, 3,
       3, 1, 0, 3, 0, 2, 0, 2, 4, 5, 4, 1, 2, 0, 3, 1, 2, 5, 3, 2, 2, 5,
       0, 0, 5, 0, 5, 3, 3, 4, 1, 4, 3, 0, 2, 2, 1, 5, 2, 6, 3, 3, 5, 1,
       5, 2, 4, 1, 3, 5, 1, 0, 5, 5, 5, 6, 2, 3, 5, 2, 4, 3, 2, 5, 0, 5,
       2, 2, 4, 2, 1, 2, 1, 3, 6, 5, 0, 5, 4, 5, 2, 5, 3, 2, 1, 0, 1,
       0, 0, 2, 3, 4, 3, 3, 0, 3, 3, 2, 5, 5, 0, 2, 6, 1, 4, 4, 5, 3, 2,
       5, 4, 4, 2, 1, 0, 0, 6, 4, 0, 0, 3, 3, 6, 5, 1, 0, 1, 6, 1, 0, 5,
       1, 4, 3, 2, 5, 3, 2, 5, 1, 1, 2, 3, 3, 5, 3, 3, 5, 2, 4, 3, 4, 2,
       0, 1, 3, 0, 6, 0, 1, 3, 4, 1, 1, 2, 1, 3, 0, 3, 4, 2, 5, 0, 5, 2,
       2, 1, 2, 5, 0, 3, 2, 3, 1, 2, 2, 2, 1, 0, 3, 5, 3, 2, 0, 3, 1, 2,
       0, 1, 1, 2, 3, 1, 5, 2, 1, 0, 3, 5, 4, 3, 5, 2, 6, 2, 5, 1, 4, 0,
       4, 0, 3, 3, 0, 0, 1, 6, 0, 5, 0, 5, 5, 5, 5, 0, 1, 3, 4, 3, 2,
       5, 1, 6, 2, 1, 6, 2, 0, 0, 0, 2, 5, 3, 2, 3, 2, 4, 3, 0, 0, 3, 6,
       6, 6, 4, 4, 3, 3, 5, 4, 3, 2, 2, 3, 3, 2, 1, 4, 2, 4, 0, 2, 5, 4,
       4, 0, 2, 5, 4, 4, 2, 2, 0, 3, 5, 6, 5, 5, 0, 3, 2, 1, 5, 3, 1, 3,
       0, 0, 1, 2, 3, 6, 0, 5, 6, 0, 2, 2, 4, 3, 4, 5, 2, 3, 0, 0, 2, 4,
       1, 3, 1, 3, 0, 4, 2, 1, 0, 1, 5, 1, 5, 3, 3, 0, 6, 0, 4, 6, 5, 5,
       4, 3, 3, 0, 6, 0, 3, 3, 6, 3, 6, 3, 4, 5, 6, 1, 4, 5, 5, 4, 0, 2,
       3, 3, 3, 2, 5, 5, 6, 1, 3, 1, 0, 3, 0, 4, 1, 2, 5, 4, 1, 4, 4, 5,
       0, 0, 2, 2, 0, 2, 1, 4, 3, 2, 6, 3, 0, 3, 2, 4, 3, 0, 6, 4, 6, 6,
       2, 5, 6, 0, 4, 5, 0, 0, 6, 3, 0, 2, 3, 2, 6, 4, 3, 6, 2, 0, 2, 2,
       0, 0, 6, 5, 3, 1, 3, 6, 5, 2, 3, 3, 1, 0, 5, 1, 5, 2, 0, 6, 0, 6,
       5, 3, 1, 6, 6, 2, 1, 1, 1, 0, 2, 0, 5, 4, 2, 1, 1, 2, 1, 2, 3, 1,
       5, 0, 0, 4, 2, 5, 3, 0, 4, 3, 6, 5, 5, 3, 4, 3, 2, 5, 0, 6, 5, 1,
       2, 1, 0, 0, 5, 1, 0, 2, 2, 1, 0, 3, 2, 5, 3, 6, 2, 1, 6, 2, 5, 1,
       2, 5, 2, 3, 3, 2, 1, 6, 2, 2, 1, 4, 4, 4, 3, 1, 0, 3, 6, 2, 1, 0,
       0, 1, 2, 2, 3, 4, 3, 0, 6, 1, 2, 2, 2, 0, 6, 0, 6, 4, 4, 0, 1, 4,
       3, 0, 6, 2, 3, 5, 5, 3, 5, 5, 0, 5, 5, 4, 4, 5, 0, 3, 3, 0, 0, 5,
       1, 5, 2, 6, 1, 0, 6, 1, 4, 2, 3, 2, 3, 3, 4, 1, 2, 4, 5, 5, 2, 4,
       1, 4, 3, 2, 4, 3, 5, 1, 3, 0, 1, 1, 2, 4, 0, 2, 2, 4, 2, 3, 1, 3,
       1, 4, 4, 6, 5, 2, 4, 6, 2, 4, 2, 3, 3, 5, 5, 2, 3, 3, 4, 1, 6, 2,
       6, 3, 4, 2, 3, 3, 3, 0, 1, 2])
```

In [8]:

```
data_with_clusters = data.copy()
data_with_clusters['Clusters'] = identified_clusters
plt.scatter(data_with_clusters['months_as_customer'], data_with_clusters['policy_annual_premium'], c=data_with_clusters['Clusters'])
```

Out[8]:

<matplotlib.collections.PathCollection at 0x1d8e8ba5550>



In [10]:

```

import seaborn as sns, numpy as np, pandas as pd, random
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
sns.set_style("whitegrid", {'axes.grid' : False})

fig = plt.figure(figsize=(6,6))

ax = Axes3D(fig)

g = ax.scatter(data_with_clusters['months_as_customer'],data_with_clusters['policy_annual_premium'],data_with_clusters['policy_de
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')

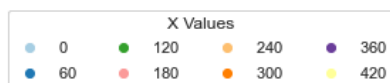
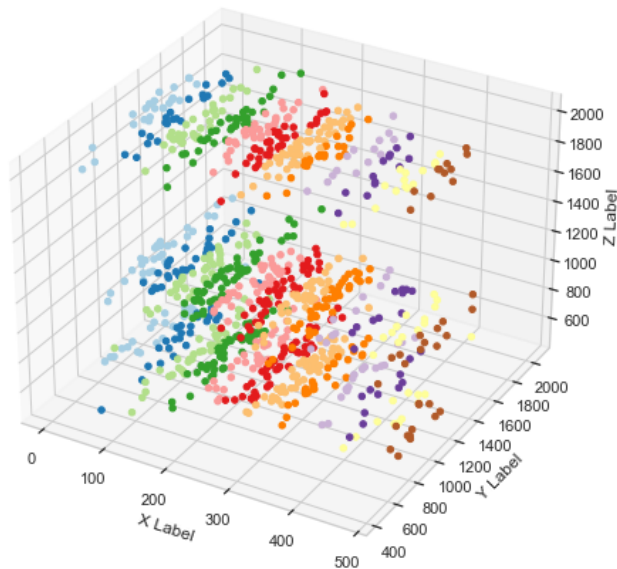
# produce a legend with the unique colors from the scatter
legend = ax.legend(*g.legend_elements(), loc="lower center", title="X Values", borderaxespad=-10, ncol=4)
ax.add_artist(legend)

plt.show()

```

C:\Users\Pranav\AppData\Local\Temp\ipykernel_8600\1669593034.py:8: MatplotlibDeprecationWarning: Axes3D(fig) adding itself to the figure is deprecated since 3.4. Pass the keyword argument auto_add_to_figure=False and use fig.add_axes(ax) to suppress this warning. The default value of auto_add_to_figure will change to False in mpl3.5 and True va

lues will no longer work in 3.6. This is consistent with other Axes classes.



In []: