In [31]:
```
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 [32]:
```
data = pd.read_csv("House_Rent_main6.csv")
data
```

Out[32]:

	BHK	Rent	Size	Area Type	Area Locality	City	Furnishing Status	Tenant Preferred	Bathroom
0	2	10000	1100	1	Bandel	1	1	Bachelors/Family	2
1	2	20000	800	1	Phool Bagan, Kankurgachi	1	2	Bachelors/Family	1
2	2	17000	1000	1	Salt Lake City Sector 2	1	2	Bachelors/Family	1
3	2	10000	800	1	Dumdum Park	1	1	Bachelors/Family	1
4	2	7500	850	2	South Dum Dum	1	1	Bachelors	1
...	...	...	...	...	...	...	...	...	...
4714	2	15000	1000	2	Bandam Kommu	6	2	Bachelors/Family	2
4715	3	29000	2000	1	Manikonda, 6	6	2	Bachelors/Family	3
4716	3	35000	1750	2	Himayath Nagar, NH 7	6	2	Bachelors/Family	3
4717	3	45000	1500	2	Gachibowli	6	2	Family	2
4718	2	15000	1000	2	Suchitra Circle	6	1	Bachelors	2

4719 rows × 9 columns

In [33]:
```
# Load your data into a pandas DataFrame
df = pd.read_csv('House_Rent_main6.csv')

# Identify the categorical variable(s) you want to encode
cat_cols = ['Area Locality']

# Perform one-hot encoding using pandas' get_dummies() function
df_encoded = pd.get_dummies(df, columns=cat_cols)

df_encoded
```

Out[33]:

	BHK	Rent	Size	Area Type	City	Furnishing Status	Tenant Preferred	Bathroom	Area Locality_ Beeramguda, Ramachandra Puram, NH 9	Lo Boc
0	2	10000	1100	1	1	1	Bachelors/Family	2	0	
1	2	20000	800	1	1	2	Bachelors/Family	1	0	
2	2	17000	1000	1	1	2	Bachelors/Family	1	0	
3	2	10000	800	1	1	1	Bachelors/Family	1	0	
4	2	7500	850	2	1	1	Bachelors	1	0	
...	...	...	...	...	...	...	...	...	...	
4714	2	15000	1000	2	6	2	Bachelors/Family	2	0	
4715	3	29000	2000	1	6	2	Bachelors/Family	3	0	
4716	3	35000	1750	2	6	2	Bachelors/Family	3	0	
4717	3	45000	1500	2	6	2	Family	2	0	
4718	2	15000	1000	2	6	1	Bachelors	2	0	

4719 rows × 2234 columns

In [34]:
```
#Replacing string values with integer
df_encoded["Tenant Preferred"].replace({"Bachelors/Family":"0", "Bachelors":"1
df_encoded
```

Out[34]:

	BHK	Rent	Size	Area Type	City	Furnishing Status	Tenant Preferred	Bathroom	Area Locality_ Beeramguda, Ramachandra Puram, NH 9	Area Locality_ in Boduppal, NH 2 2
0	2	10000	1100	1	1	1	0	2	0	0
1	2	20000	800	1	1	2	0	1	0	0
2	2	17000	1000	1	1	2	0	1	0	0
3	2	10000	800	1	1	1	0	1	0	0
4	2	7500	850	2	1	1	1	1	0	0
...	...	...	...	...	...	...	...	...	...	...
4714	2	15000	1000	2	6	2	0	2	0	0
4715	3	29000	2000	1	6	2	0	3	0	0
4716	3	35000	1750	2	6	2	0	3	0	0
4717	3	45000	1500	2	6	2	2	2	0	0
4718	2	15000	1000	2	6	1	1	2	0	0

4719 rows × 2234 columns

In [60]:
```
# Using K-Means Clustering
x = df_encoded.iloc[:,0:3] # 1st for rows and second for columns
x
```

Out[60]:

	BHK	Rent	Size
0	2	10000	1100
1	2	20000	800
2	2	17000	1000
3	2	10000	800
4	2	7500	850
...	...	...	...
4714	2	15000	1000
4715	3	29000	2000
4716	3	35000	1750
4717	3	45000	1500
4718	2	15000	1000

4719 rows × 3 columns

In [61]:
```
kmeans = KMeans(3) # Creating three clusters
kmeans.fit(x)
```

```
C:\Users\Asus\AppData\Local\Programs\Python\Python310\lib\site-packages\sklea
rn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will
change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to supp
ress the warning
  warnings.warn(
```

Out[61]:

KMeans

KMeans(n_clusters=3)

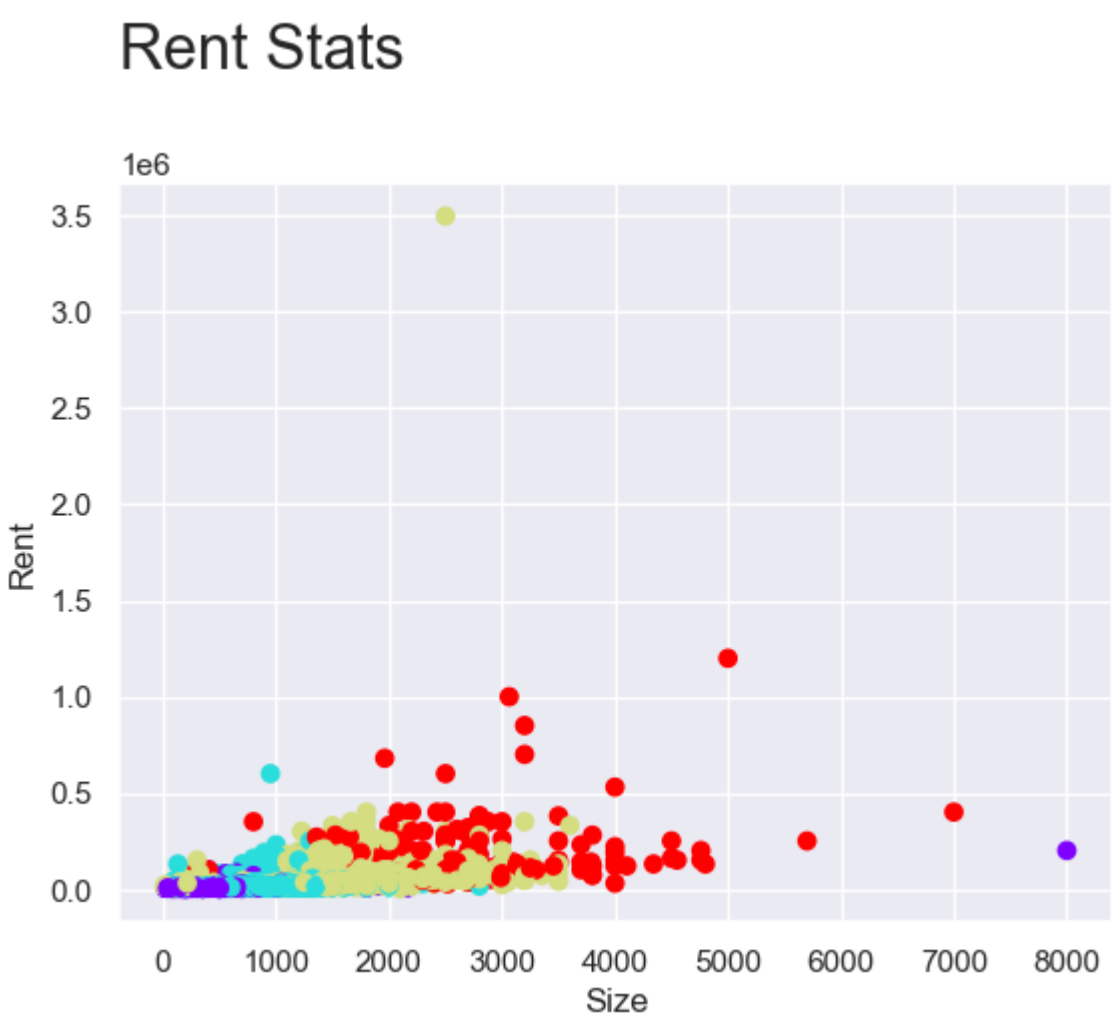
In [62]:
```
identified_clusters = kmeans.fit_predict(x)
identified_clusters
```

```
C:\Users\Asus\AppData\Local\Programs\Python\Python310\lib\site-packages\sklea
rn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will
change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to supp
ress the warning
  warnings.warn(
```

Out[62]:
```
array([0, 0, 0, ..., 0, 0, 0])
```

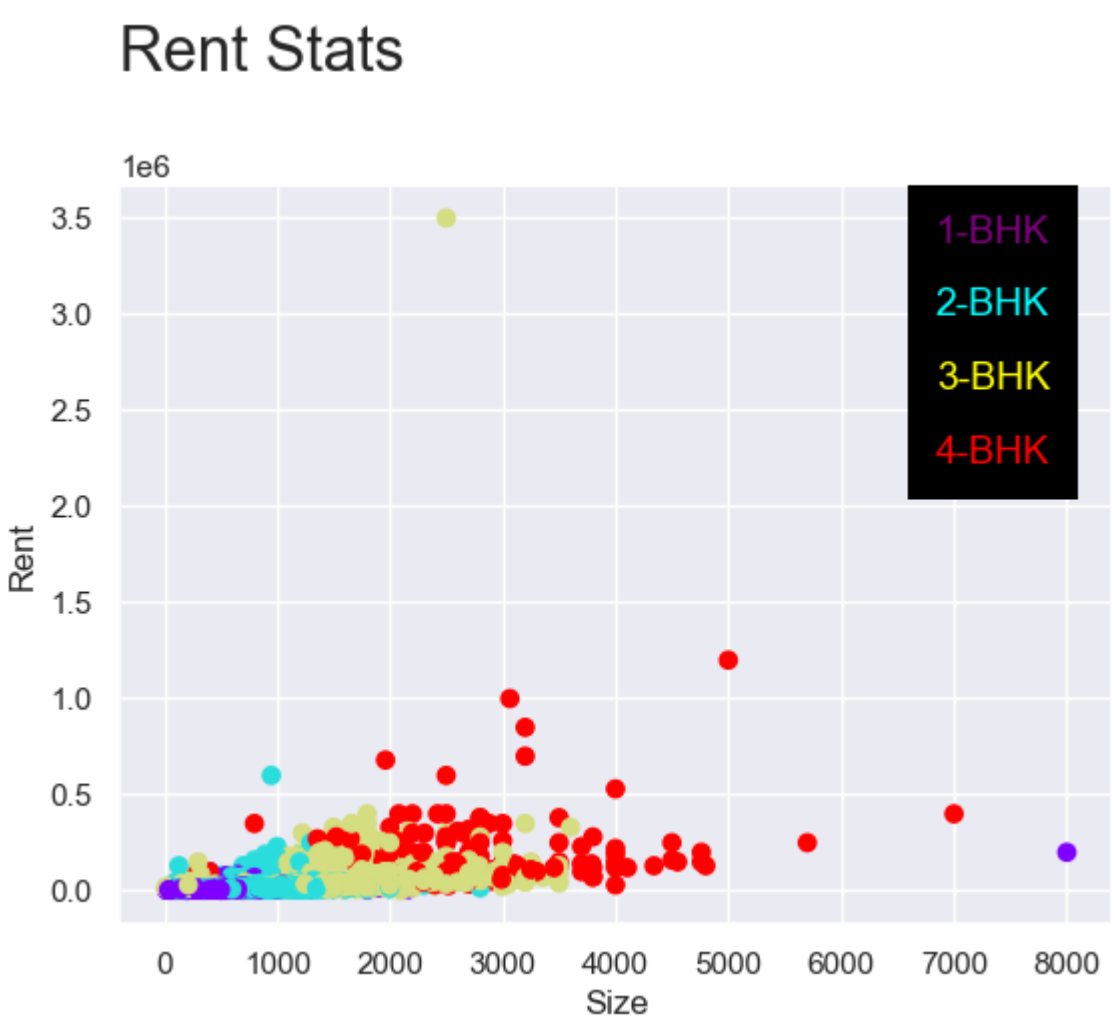
In [92]:
```
data_with_clusters = df_encoded.copy()
data_with_clusters['Clusters'] = identified_clusters
plt.scatter(data_with_clusters['Size'],data_with_clusters['Rent'],c=data_with_
plt.title('Rent Stats\n', loc='left', fontsize=22)
plt.xlabel('Size')
plt.ylabel('Rent')
```

Out[92]:
```
Text(0, 0.5, 'Rent')
```



In [91]:
```
data_with_clusters = df_encoded.copy()
data_with_clusters['Clusters'] = identified_clusters
plt.scatter(data_with_clusters['Size'],data_with_clusters['Rent'],c=data_with_
plt.title('Rent Stats\n', loc='left', fontsize=22)
plt.xlabel('Size')
plt.ylabel('Rent')

import matplotlib.patches as mpatches

# create purple circle
purple_circle = mpatches.Circle((0.8, 0.8), radius=0.1, facecolor='purple')

# add circle to plot
plt.gca().add_patch(purple_circle)

# add label text
plt.text(0.85, 0.95, '1-BHK', color='purple',transform=ax.transAxes, fontsize=
        verticalalignment='top', horizontalalignment='right', ha='center', va=

#--


# create purple circle
cyan_circle = mpatches.Circle((0.8, 0.8), radius=0.1, facecolor='cyan')

# add circle to plot
plt.gca().add_patch(cyan_circle)

# add label text
plt.text(0.85, 0.85, '2-BHK', color='cyan',transform=ax.transAxes, fontsize=14
        verticalalignment='top', horizontalalignment='right', ha='center', va=
#--


# create purple circle
yellow_circle = mpatches.Circle((0.8, 0.8), radius=0.1, facecolor='yellow')

# add circle to plot
plt.gca().add_patch(yellow_circle)

# add label text
plt.text(0.85, 0.75, '3-BHK', color='yellow',transform=ax.transAxes, fontsize=
        verticalalignment='top', horizontalalignment='right', ha='center', va=
#--
```

Out[91]:
```
Text(0.85, 0.65, '4-BHK')
```


Rent Stats
1e6
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Rent
1-BHK
2-BHK
3-BHK
4-BHK
0
1000
2000
3000
4000
5000
6000
7000
8000
Size

In [ ]: