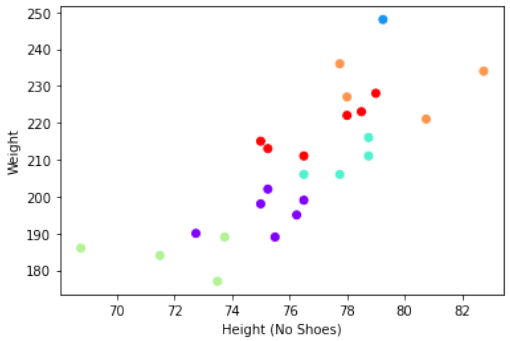
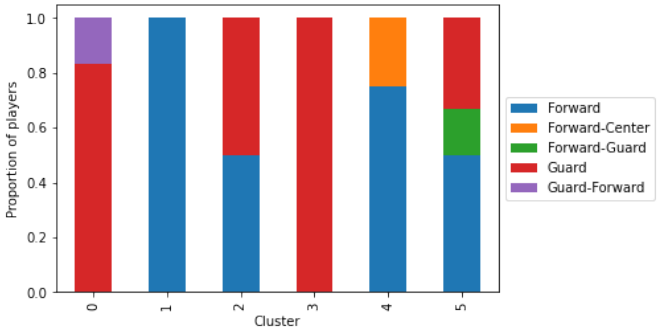
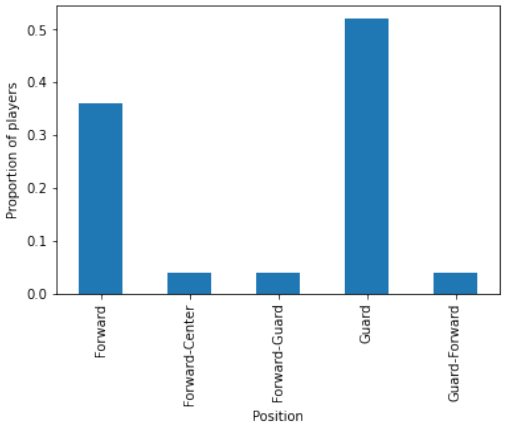
In [2]:
```
import pandas as pd
from sklearn.cluster import KMeans
from scipy.stats import chi2_contingency
import matplotlib.pyplot as plt

# load the dataset
df = pd.read_excel("nbadata_cleaned.xlsx")
df["Position_num"] = df["POSITION"].map({"Guard": 1, "Guard-Forward": 2, "Forward": 3, "Forward-Guard": 4, "Forward-Center"

# drop rows with missing values
df = df.dropna()

# select the columns to use for clustering
cols = ['Height (No Shoes)', 'Height (With Shoes)', 'Wingspan', 'Standing reach', 'Vertical (Max)', 'Vertical (Max Reach)',

# apply k-means clustering
k = 6
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(df[cols])

# get the cluster labels and add them to the dataframe
df['Cluster'] = kmeans.labels_

# print the number of players in each cluster
print(df['Cluster'].value_counts())

# plot the clusters
plt.scatter(df['Height (No Shoes)'], df['Weight'], c=kmeans.labels_, cmap='rainbow')
plt.xlabel('Height (No Shoes)')
plt.ylabel('Weight')
plt.show()

# get the player names for each cluster
for i in range(k):
    print(f"Cluster {i}:")
    print(df[df['Cluster'] == i]['Player'].values)




# create a frequency table to analyze the overlap between clusters and positions
freq_table = pd.crosstab(df['POSITION'], df['Cluster'], margins=True)
print(freq_table)




# perform a chi-square test to test for significant differences between clusters and positions
chi2, pval, dof, exp_freq = chi2_contingency(freq_table.iloc[:-1,:-1])
print("Chi-square test statistic:", chi2)
print("P-value:", pval)


# calculate the proportion of players in each position for each cluster
cluster_props = df.groupby(['Cluster', 'POSITION'])['Player'].count() / df.groupby('Cluster')['Player'].count()

# calculate the proportion of players in each position in the original dataset
pos_props = df.groupby('POSITION')['Player'].count() / len(df)

# plot the proportion of players in each position for each cluster
cluster_props.unstack().plot(kind='bar', stacked=True)
plt.legend(loc='center left', bbox_to_anchor=(1.0, 0.5))
plt.xlabel('Cluster')
plt.ylabel('Proportion of players')
plt.show()

# plot the proportion of players in each position in the original dataset
pos_props.plot(kind='bar')
plt.xlabel('Position')
plt.ylabel('Proportion of players')
plt.show()
```

```
0    6
5    6
2    4
4    4
3    4
1    1
Name: Cluster, dtype: int64
```




```
Cluster 0:
['Jordan Crawford' 'Marshon Brooks' 'Bradley Beal' 'Tim Hardaway Jr'
 'Ben McLemore' 'Terry Rozier']
Cluster 1:
['Derrick Williams']
Cluster 2:
['Gordon Hayward' 'Klay Thompson' 'Khris Middleton' 'Devin Booker']
Cluster 3:
['Isaiah Thomas' 'Brandon Knight' 'Kemba Walker' 'Damian Lillard']
Cluster 4:
['Kawhi Leonard' 'Chandler Parsons' 'Draymond Green' 'Kelly Olynyk']
Cluster 5:
['Tobias Harris' 'Jimmy Butler' 'Harrison Barnes' 'Victor Oladipo'
 'Glen Rice' 'Norman Powell']
Cluster         0  1  2  3  4  5  All
POSITION
Forward         0  1  2  0  3  3    9
Forward-Center  0  0  0  0  1  0    1
Forward-Guard   0  0  0  0  0  1    1
Guard           5  0  2  4  0  2   13
Guard-Forward   1  0  0  0  0  0    1
All             6  1  4  4  4  6   25
Chi-square test statistic: 24.465811965811966
P-value: 0.22263642626103175
```





In [ ]: