In [39]:

```python
import os
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import IPython.display as display
from sklearn.cluster import AgglomerativeClustering
from sklearn import datasets
```

In [34]:

```python
iris_data=datasets.load_iris()
X=iris_data.data[:, :2]
y=iris_data.target
target_names = iris_data.target_names
cluster=AgglomerativeClustering(linkage="ward", n_clusters=3)
cluster.fit(X);
```

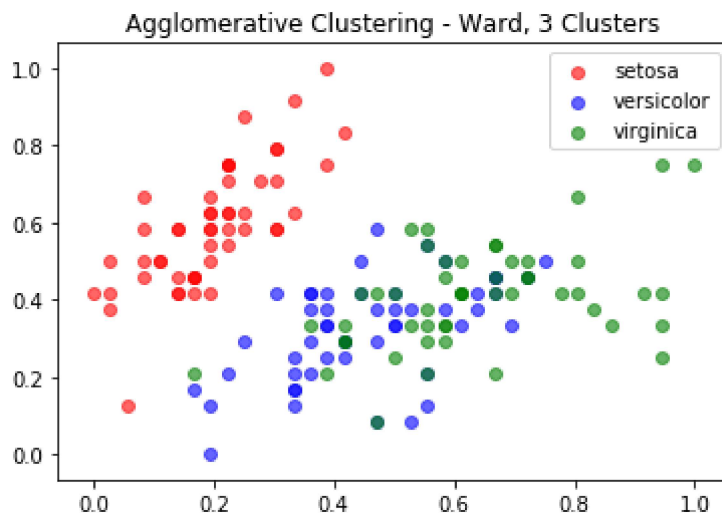Loading [MathJax]/extensions/Safe.js

In [52]:

```python
from sklearn import preprocessing
from scipy.cluster.hierarchy import dendrogram, linkage
X_plot = preprocessing.MinMaxScaler().fit_transform(X)

plt.figure()
colors = 'rbg'
lw = 1

for color, i, target_name in zip(colors, [0, 1, 2], target_names):
    plt.scatter(X_plot[y == i, 0], X_plot[y == i, 1], color=color, alpha=.6, lw=lw,
                label=target_name)
plt.legend(loc='best', shadow=False, scatterpoints=1)
plt.title('Agglomerative Clustering - Ward, 3 Clusters')

figure=plt.figure(figsize=(7,5))
Z=linkage(X,'ward')
dendrogram(
    Z,
    truncate_mode='lastp',
    p=24,
    leaf_rotation=90.,
    leaf_font_size=12.,
    show_contracted=True,   #to get a distribution impression in truncated branches
)
plt.title('Hierarchical Clustering Dendogram (Ward, aggrogated)')
plt.xlabel('sample index or (cluster size)')
plt.ylabel('distance')
plt.show()
```
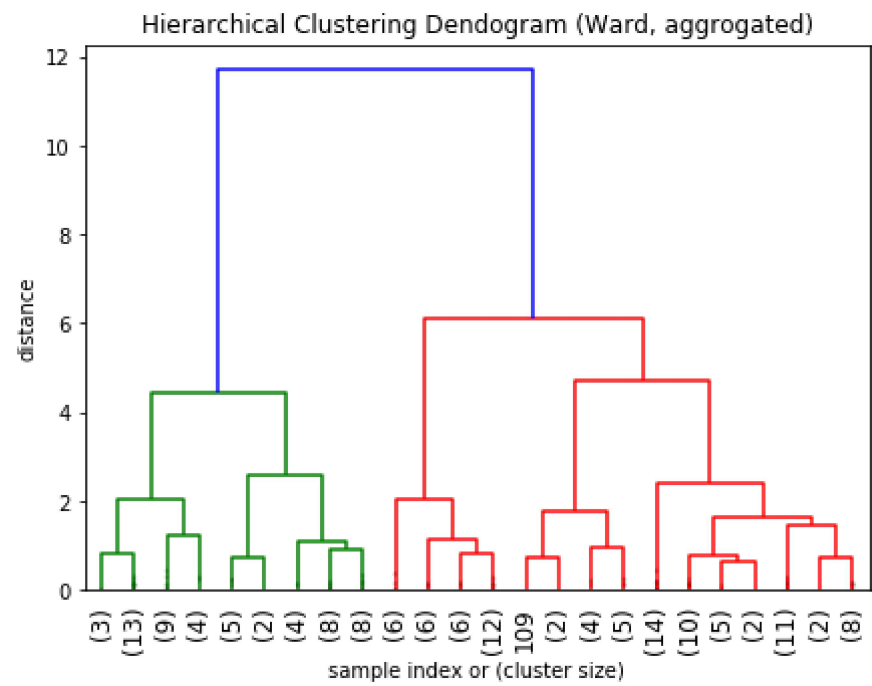


Loading [MathJax]/extensions/Safe.js

## Hierarchical Clustering Dendogram (Ward, aggrogated)



In [ ]: