In [76]:

```
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.cluster import KMeans

# Load the dataset which was used throughout the whole course
data = pd.read_excel('finally_clean_data_for_plotting.xlsx')

# Upper-case all DataFrame column names
data.columns = map(str.upper, data.columns)

# Data Cleaning
cleaned_data = data.dropna()

# Show the data frame
print(cleaned_data)
```

```
       AGE  SEX  HOUSEHOLDINCOME  HOWOFTENWINE  NOOFWINES     INCOME_CATEGORY
0       34    2               12            10          1  $50,000 to $59,999
1       84    2                7             6          1  $20,000 to $24,999
2       29    2               13            10          1  $60,000 to $69,999
3       68    2                6             5          1  $15,000 to $19,999
4       54    2               11             9          1  $40,000 to $49,999
...    ...  ...              ...           ...        ...                 ...
14556   18    2                1             9          1    Less than $5,000
14557   18    1                1            10          1    Less than $5,000
14558   51    1                6             6          1  $15,000 to $19,999
14559   21    1                1            10          2    Less than $5,000
14560   18    2                1            10          1    Less than $5,000

[14561 rows x 6 columns]
```

The dataset focusses on the correlations between age, sex, household-income (already pre-categorized according to the NESARC codebook) and wine drinking frequency with the "noofwines" variable, which represents the consumed amount of wine per occasion.

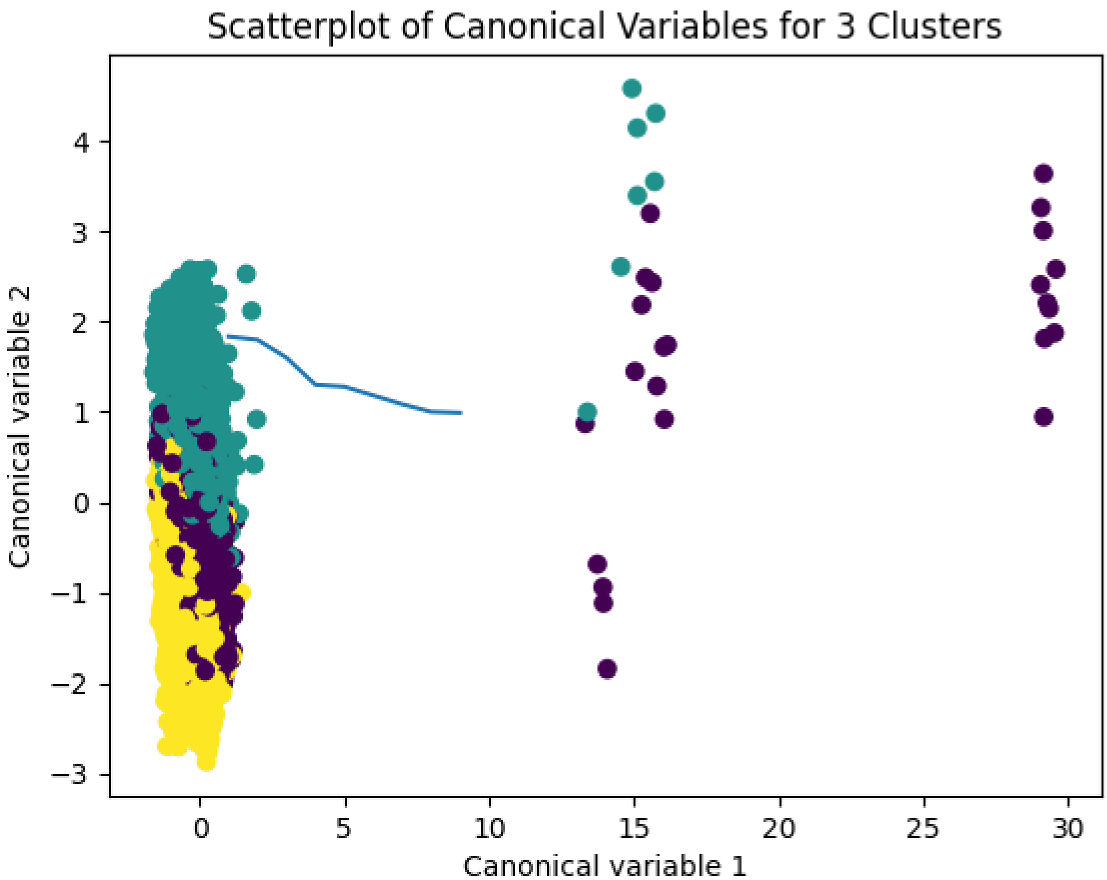
In [77]:

```
cluster = cleaned_data[['AGE', 'SEX', 'HOWOFTENWINE', 'HOUSEHOLDINCOME', 'NOOFWINES']
cluster.describe()

clustervar=cluster.copy()
clustervar['AGE'] = preprocessing.scale(clustervar['AGE'].astype('float64'))
clustervar['SEX'] = preprocessing.scale(clustervar['SEX'].astype('float64'))
clustervar['HOWOFTENWINE'] = preprocessing.scale(clustervar['HOWOFTENWINE'].astype('
clustervar['HOUSEHOLDINCOME'] = preprocessing.scale(clustervar['HOUSEHOLDINCOME'].ast
clustervar['NOOFWINES'] = preprocessing.scale(clustervar['NOOFWINES'].astype('float64

# split data into train and test sets
clus_train, clus_test = train_test_split(clustervar, test_size = .3, random_state = 
```

In the following, the k-means cluster analysis is being performed, again, straight forward following the example of the course.

In [80]:

```
# k-means cluster analysis for 1-9 clusters
from scipy.spatial.distance import cdist
clusters = range(1,10)
meandist = []

for k in clusters:
    model = KMeans(n_clusters = k)
    model.fit(clus_train)
    clusassign = model.predict(clus_train)
    meandist.append(sum(np.min(cdist(clus_train, model.cluster_centers_, 'euclidean')
    / clus_train.shape[0])

plt.plot(clusters, meandist)
plt.xlabel('Number of clusters')
plt.ylabel('Average distance')
plt.title('Selecting k with the Elbow Method')

# Interpret 3 cluster solution
model3 = KMeans(n_clusters = 3)
model3.fit(clus_train)
clusassign = model3.predict(clus_train)

# plot clusters
from sklearn.decomposition import PCA

pca_2 = PCA(2)
plot_columns = pca_2.fit_transform(clus_train)
plt.scatter(x = plot_columns[:,0], y = plot_columns[:,1], c = model3.labels_,)
plt.xlabel('Canonical variable 1')
plt.ylabel('Canonical variable 2')
plt.title('Scatterplot of Canonical Variables for 3 Clusters')
plt.show()
```



Observations and Interpretations:

The points are primarily clustered along the y-axis, which may indicate that the variance in the data is predominantly in one dimension (the y-axis in the PCA plot). This could mean that one or more features are dominating the clustering process. It might also suggest that the features used for clustering are not well-scaled or that they have different ranges, which can lead to poor clustering results.

The vertical lines observed for the purple and turqouise cluster at 15 and 30 of the x-axis could indicate that there are outliers or that certain data points are very similar in one dimension but vary significantly in the another.

As in the course example: Multiple steps to merge cluster assignment with clustering variables to examine cluster variable means by cluster

In [71]:

```
# Create a unique identifier variable from the index for the
# Cluster training data to merge with the cluster assignment variable
clus_train.reset_index(level = 0, inplace = True)

# Create a list that has the new index variable
cluslist = list(clus_train['index'])

# Create a list of cluster assignments
labels = list(model3.labels_)

# Combine index variable list with cluster assignment list into a dictionary
newlist = dict(zip(cluslist, labels))
newlist

# Convert newlist dictionary to a dataframe
newclus = DataFrame.from_dict(newlist, orient = 'index')
newclus

# Rename the cluster assignment column
newclus.columns = ['cluster']

# Now do the same for the cluster assignment variable
# Create a unique identifier variable from the index for the
# cluster assignment data frame
# to merge with cluster training data
newclus.reset_index(level = 0, inplace = True)

# Merge the cluster assignment dataframe with the cluster training variable dataframe
# by the index variable
merged_train = pd.merge(clus_train, newclus, on = 'index')
merged_train.head(n = 100)

# cluster frequencies
merged_train.cluster.value_counts()
```

Out[71]:

```
cluster
0    5927
1    4240
2      25
Name: count, dtype: int64
```

In [72]:

```
Calculate clustering variable means by cluster
clustergrp = merged_train.groupby('cluster').mean()
print ("Clustering variable means by cluster")
print(clustergrp)


# Validate clusters in training data by examining cluster differences in GPA using AN
# first have to merge GPA with clustering variables and cluster assignment data
noofwines_data = cleaned_data['NOOFWINES']

# Perform the merge
merged_train_all = pd.merge(noofwines_train1, merged_train, on='index')

# Drop one of the NOOFWINES columns if they exist
# (I have experienced some problems with the python df merge and therefor
# introduced this piece of code to fix it)
if 'NOOFWINES_x' in merged_train_all.columns:
    merged_train_all.drop(columns=['NOOFWINES_x'], inplace=True)
if 'NOOFWINES_y' in merged_train_all.columns:
    merged_train_all.rename(columns={'NOOFWINES_y': 'NOOFWINES'}, inplace=True)

# Now create sub1 using the correct NOOFWINES column
sub1 = merged_train_all[['NOOFWINES', 'cluster']].dropna()

import statsmodels.formula.api as smf
import statsmodels.stats.multicomp as multi

gpamod = smf.ols(formula = 'NOOFWINES ~ C(cluster)', data = sub1).fit()
print (gpamod.summary())

print ('Means for NOOFWINES by cluster')
m1= sub1.groupby('cluster').mean()
print (m1)

print ('Standard Deviations for NOOFWINES by cluster')
m2= sub1.groupby('cluster').std()
print (m2)

mc1 = multi.MultiComparison(sub1['NOOFWINES'], sub1['cluster'])
res1 = mc1.tukeyhsd()
print(res1.summary())
```

```
Clustering variable means by cluster
                index       AGE       SEX  HOWOFTENWINE  HOUSEHOLDINCOME  \
cluster
0        7308.844609 -0.009714  0.847893     -0.000656        -0.091330
1        7262.246698  0.017298 -1.179393     -0.039147         0.135703
2        7901.200000  0.192379 -0.044113      8.311003        -0.511209

         NOOFWINES
cluster
0        -0.056375
1        -0.031799
2        21.332474
                            OLS Regression Results
==============================================================================
Dep. Variable:              NOOFWINES   R-squared:                       0.975
```

```
Model:                            OLS   Adj. R-squared:                  0.975
Method:                 Least Squares   F-statistic:                 1.956e+05
Date:                Thu, 03 Apr 2025   Prob (F-statistic):               0.00
Time:                        14:56:52   Log-Likelihood:                 3557.8
No. Observations:               10192   AIC:                            -7110.
Df Residuals:                   10189   BIC:                            -7088.
Df Model:                           2
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept         -0.0564      0.002    -25.426      0.000      -0.061      -0.052
C(cluster)[T.1]    0.0246      0.003      7.158      0.000       0.018       0.031
C(cluster)[T.2]   21.3888      0.034    625.202      0.000      21.322      21.456
==============================================================================
Omnibus:                     6623.514   Durbin-Watson:                   1.970
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           132207.239
Skew:                           2.811   Prob(JB):                         0.00
Kurtosis:                      19.724   Cond. No.                         22.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
ified.
Means for NOOFWINES by cluster
         NOOFWINES
cluster
0        -0.056375
1        -0.031799
2        21.332474
Standard Deviations for NOOFWINES by cluster
         NOOFWINES
cluster
0         0.157386
1         0.188167
2         0.000000
Multiple Comparison of Means - Tukey HSD, FWER=0.05
===================================================
group1 group2 meandiff p-adj  lower   upper  reject
---------------------------------------------------
     0      1   0.0246   0.0  0.0165  0.0326   True
     0      2  21.3888   0.0 21.3087  21.469   True
     1      2  21.3643   0.0  21.284 21.4445   True
---------------------------------------------------
```

Interpretation of the results:

Clustering Variable Means by Cluster:

```
Cluster 0:
    NOOFWINES: -0.056375
Cluster 1:
    NOOFWINES: -0.031799
Cluster 2:
    NOOFWINES: 21.332474
```

The means suggest that Cluster 2 has a significantly higher average value for the wines consumed per occasion (NOOFWINES) compared to Clusters 0 and 1.

OLS Regression Results:

R-squared: 0.975 indicates that approximately 97.5% of the variance in NOOFWINES can be explained by the cluster assignments, which is a very good value, compared to the ones of the last assignments.

- The intercept is -0.0564, which is the expected value of NOOFWINES for Cluster 0.
- The coefficient for Cluster 1 is 0.0246, indicating that being in Cluster 1 increases the expected NOOFWINES by about 0.0246 compared to Cluster 0.
- The coefficient for Cluster 2 is 21.3888, indicating a significant increase in expected NOOFWINES compared to Cluster 0.

Tukey HSD Results

All comparisons between clusters (0 vs 1, 0 vs 2, and 1 vs 2) are statistically significant ($p < 0.05$), meaning there are significant differences in NOOFWINES between these clusters.

Summary:

Cluster Analysis: Three clusters with distinct characteristics in terms of the NOOFWINES variable were identified.

Statistical Significance: The analysis shows that the differences in NOOFWINES between the clusters are statistically significant, which suggests that the clustering is meaningful.