K-Means Clustering

April 15, 2021

1 Helper Functions

[1]:
```python
# (c) 2014 Reid Johnson
#
# Modified from:
# (c) 2013 Mikael Vejdemo-Johansson
# BSD License
#
# SciPy function to compute the gap statistic for evaluating k-means clustering.
#
# The gap statistic is defined by Tibshirani, Walther, Hastie in:
#  Estimating the number of clusters in a data set via the gap statistic
#  J. R. Statist. Soc. B (2001) 63, Part 2, pp 411-423
import numpy as np
from numpy.linalg import LinAlgError
import scipy as sp
import sklearn.cluster
from scipy.spatial.distance import cdist, pdist
import pylab as pl
import scipy.cluster.vq
import scipy.spatial.distance
import scipy.stats
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

import pylab as pl

dst = sp.spatial.distance.euclidean

def gap_statistics(data, refs=None, nrefs=20, ks=range(1,11)):
    """Computes the gap statistics for an nxm dataset.

    The gap statistic measures the difference between within-cluster dispersion␣
↪on an input
    dataset and that expected under an appropriate reference null distribution.
```

```
    Computation of the gap statistic, then, requires a series of reference␣
↪(null) distributions.
    One may either input a precomputed set of reference distributions (via the␣
↪parameter refs)
    or specify the number of reference distributions (via the parameter nrefs)␣
↪for automatic
    generation of uniform distributions within the bounding box of the dataset␣
↪(data).

    Each computation of the gap statistic requires the clustering of the input␣
↪dataset and of
    several reference distributions. To identify the optimal number of clusters␣
↪k, the gap
    statistic is computed over a range of possible values of k (via the␣
↪parameter ks).

    For each value of k, within-cluster dispersion is calculated for the input␣
↪dataset and each
    reference distribution. The calculation of the within-cluster dispersion␣
↪for the reference
    distributions will have a degree of variation, which we measure by standard␣
↪deviation or
    standard error.

    The estimated optimal number of clusters, then, is defined as the smallest␣
↪value k such that
    gap_k is greater than or equal to the sum of gap_k+1 minus the expected␣
↪error err_k+1.

    Args:
      data ((n,m) SciPy array): The dataset on which to compute the gap␣
↪statistics.
      refs ((n,m,k) SciPy array, optional): A precomputed set of reference␣
↪distributions.
        Defaults to None.
      nrefs (int, optional): The number of reference distributions for␣
↪automatic generation.
        Defaults to 20.
      ks (list, optional): The list of values k for which to compute the gap␣
↪statistics.
        Defaults to range(1,11), which creates a list of values from 1 to 10.

    Returns:
      gaps: an array of gap statistics computed for each k.
      errs: an array of standard errors (se), with one corresponding to each␣
↪gap computation.
```

```
    difs: an array of differences between each gap_k and the sum of gap_k+1␣
↪minus err_k+1.

  """
  shape = data.shape

  if refs==None:
      tops = data.max(axis=0) # maxima along the first axis (rows)
      bots = data.min(axis=0) # minima along the first axis (rows)
      dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input␣
↪dataset

      # Generate nrefs uniform distributions each in the half-open interval␣
↪[0.0, 1.0)
      rands = sp.random.random_sample(size=(shape[0],shape[1], nrefs))

      # Adjust each of the uniform distributions to the bounding box of the␣
↪input dataset
      for i in range(nrefs):
          rands[:,:,i] = rands[:,:,i]*dists+bots
  else:
      rands = refs

  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length␣
↪ks-1)

  for (i,k) in enumerate(ks): # iterate over the range of k values
      # Cluster the input dataset via k-means clustering using the current␣
↪value of k
      try:
          (kmc,kml) = sp.cluster.vq.kmeans2(data, k)
      except LinAlgError:
          kmeans = sklearn.cluster.KMeans(n_clusters=k).fit(data)
          (kmc, kml) = kmeans.cluster_centers_, kmeans.labels_

      # Generate within-dispersion measure for the clustering of the input␣
↪dataset
      disp = sum([dst(data[m,:],kmc[kml[m],:]) for m in range(shape[0])])

      # Generate within-dispersion measures for the clusterings of the␣
↪reference datasets
      refdisps = sp.zeros((rands.shape[2],))
      for j in range(rands.shape[2]):
```

```
            # Cluster the reference dataset via k-means clustering using the␣
↪current value of k
            try:
                (kmc,kml) = sp.cluster.vq.kmeans2(rands[:,:,j], k)
            except LinAlgError:
                kmeans = sklearn.cluster.KMeans(n_clusters=k).fit(rands[:,:,j])
                (kmc, kml) = kmeans.cluster_centers_, kmeans.labels_

            refdisps[j] = sum([dst(rands[m,:,j],kmc[kml[m],:]) for m in␣
↪range(shape[0])])

        # Compute the (estimated) gap statistic for k
        gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))

        # Compute the expected error for k
        errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
                              for refdisp in refdisps)/float(nrefs)) * sp.
↪sqrt(1+1/nrefs)

    # Compute the difference between gap_k and the sum of gap_k+1 minus err_k+1
    difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in␣
↪range(len(gaps)-1)])

    #print "Gaps: " + str(gaps)
    #print "Errs: " + str(errs)
    #print "Difs: " + str(difs)

    return gaps, errs, difs

def plot_gap_statistics(gaps, errs, difs):
    """Generates and shows plots for the gap statistics.

    A figure with two subplots is generated. The first subplot is an errorbar␣
↪plot of the
    estimated gap statistics computed for each value of k. The second subplot␣
↪is a barplot
    of the differences in the computed gap statistics.

    Args:
      gaps (SciPy array): An array of gap statistics, one computed for each k.
      errs (SciPy array): An array of standard errors (se), with one␣
↪corresponding to each gap
        computation.
      difs (SciPy array): An array of differences between each gap_k and the␣
↪sum of gap_k+1
        minus err_k+1.
```

```
"""
# Create a figure
fig = pl.figure(figsize=(16, 4))

pl.subplots_adjust(wspace=0.35) # adjust the distance between figures

# Subplot 1
ax = fig.add_subplot(121)
ind = range(1,len(gaps)+1) # the x values for the gaps

# Create an errorbar plot
rects = ax.errorbar(ind, gaps, yerr=errs, xerr=None, linewidth=1.0)

# Add figure labels and ticks
ax.set_title('Clustering Gap Statistics', fontsize=16)
ax.set_xlabel('Number of clusters k', fontsize=14)
ax.set_ylabel('Gap Statistic', fontsize=14)
ax.set_xticks(ind)

# Add figure bounds
ax.set_ylim(0, max(gaps+errs)*1.1)
ax.set_xlim(0, len(gaps)+1.0)

# Subplot 2
ax = fig.add_subplot(122)
ind = range(1,len(difs)+1) # the x values for the difs

max_gap = None
if len(np.where(difs > 0)[0]) > 0:
    max_gap = np.where(difs > 0)[0][0] + 1 # the k with the first positive dif

# Create a bar plot
ax.bar(ind, difs, alpha=0.5, color='g', align='center')

# Add figure labels and ticks
if max_gap:
    ax.set_title('Clustering Gap Differences\n(k=%d Estimated as Optimal)' % (max_gap), \
                 fontsize=16)
else:
    ax.set_title('Clustering Gap Differences\n', fontsize=16)
ax.set_xlabel('Number of clusters k', fontsize=14)
ax.set_ylabel('Gap Difference', fontsize=14)
ax.xaxis.set_ticks(range(1,len(difs)+1))
```

```
    # Add figure bounds
    ax.set_ylim(min(difs)*1.2, max(difs)*1.2)
    ax.set_xlim(0, len(difs)+1.0)

    # Show the figure
    pl.show()

# (c) 2014 Reid Johnson
# BSD License
#
# Function to compute the sum of squared distance (SSQ) for evaluating k-means␣
↪clustering.

def ssq_statistics(data, ks=range(1,11), ssq_norm=True):
    """Computes the sum of squares for an nxm dataset.

    The sum of squares (SSQ) is a measure of within-cluster variation that␣
↪measures the sum of
    squared distances from cluster prototypes.

    Each computation of the SSQ requires the clustering of the input dataset.␣
↪To identify the
    optimal number of clusters k, the SSQ is computed over a range of possible␣
↪values of k
    (via the parameter ks). For each value of k, within-cluster dispersion is␣
↪calculated for the
    input dataset.

    The estimated optimal number of clusters, then, is defined as the value of␣
↪k prior to an
    "elbow" point in the plot of SSQ values.

    Args:
      data ((n,m) SciPy array): The dataset on which to compute the gap␣
↪statistics.
      ks (list, optional): The list of values k for which to compute the gap␣
↪statistics.
        Defaults to range(1,11), which creates a list of values from 1 to 10.

    Returns:
      ssqs: an array of SSQs, one computed for each k.

    """
    ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```

```
    #n_samples, n_features = data.shape # the number of rows (samples) and columns (features)
    #if n_samples >= 2500:
    #    # Generate a small sub-sample of the data
    #    data_sample = shuffle(data, random_state=0)[:1000]
    #else:
    #    data_sample = data

    for (i,k) in enumerate(ks): # iterate over the range of k values
        # Fit the model on the data
        kmeans = sklearn.cluster.KMeans(n_clusters=k, random_state=0).fit(data)

        # Predict on the data (k-means) and get labels
        #labels = kmeans.predict(data)

        if ssq_norm:
            dist = np.min(cdist(data, kmeans.cluster_centers_, 'euclidean'), axis=1)

            tot_withinss = sum(dist**2) # Total within-cluster sum of squares
            totss = sum(pdist(data)**2) / data.shape[0] # The total sum of squares
            betweenss = totss - tot_withinss # The between-cluster sum of squares
            ssqs[i] = betweenss/totss*100
        else:
            # The sum of squared error (SSQ) for k
            ssqs[i] = kmeans.inertia_

    return ssqs

def plot_ssq_statistics(ssqs):
    """Generates and shows plots for the sum of squares (SSQ).

    A figure with one plot is generated. The plot is a bar plot of the SSQ computed for each
    value of k.

    Args:
      ssqs (SciPy array): An array of SSQs, one computed for each k.

    """
    # Create a figure
    fig = pl.figure(figsize=(6.75, 4))

    ind = range(1,len(ssqs)+1) # the x values for the ssqs
    width = 0.5 # the width of the bars
```

```
    # Create a bar plot
    #rects = pl.bar(ind, ssqs, width)
    pl.plot(ind, ssqs)

    # Add figure labels and ticks
    pl.title('Clustering Sum of Squared Distances', fontsize=16)
    pl.xlabel('Number of clusters k', fontsize=14)
    pl.ylabel('Sum of Squared Distance (SSQ)', fontsize=14)
    pl.xticks(ind)

    # Add text labels
    #for rect in rects:
    #    height = rect.get_height()
    #    pl.text(rect.get_x()+rect.get_width()/2., 1.05*height, '%d' %␣
↪int(height), \
    #            ha='center', va='bottom')

    # Add figure bounds
    pl.ylim(0, max(ssqs)*1.2)
    pl.xlim(0, len(ssqs)+1.0)

    pl.show()
```

[2]:
```
import pandas as pd
import numpy as np
# load the data set
df = pd.read_csv('shopping-data.csv')
d = df[["Annual Income (k$)", "Spending Score (1-100)"]]
data = d.to_numpy(dtype ='float64')
print(data)
```

```
[[ 15.  39.]
 [ 15.  81.]
 [ 16.   6.]
 [ 16.  77.]
 [ 17.  40.]
 [ 17.  76.]
 [ 18.   6.]
 [ 18.  94.]
 [ 19.   3.]
 [ 19.  72.]
 [ 19.  14.]
 [ 19.  99.]
 [ 20.  15.]
 [ 20.  77.]
 [ 20.  13.]
```

```
[ 20.  79.]
[ 21.  35.]
[ 21.  66.]
[ 23.  29.]
[ 23.  98.]
[ 24.  35.]
[ 24.  73.]
[ 25.   5.]
[ 25.  73.]
[ 28.  14.]
[ 28.  82.]
[ 28.  32.]
[ 28.  61.]
[ 29.  31.]
[ 29.  87.]
[ 30.   4.]
[ 30.  73.]
[ 33.   4.]
[ 33.  92.]
[ 33.  14.]
[ 33.  81.]
[ 34.  17.]
[ 34.  73.]
[ 37.  26.]
[ 37.  75.]
[ 38.  35.]
[ 38.  92.]
[ 39.  36.]
[ 39.  61.]
[ 39.  28.]
[ 39.  65.]
[ 40.  55.]
[ 40.  47.]
[ 40.  42.]
[ 40.  42.]
[ 42.  52.]
[ 42.  60.]
[ 43.  54.]
[ 43.  60.]
[ 43.  45.]
[ 43.  41.]
[ 44.  50.]
[ 44.  46.]
[ 46.  51.]
[ 46.  46.]
[ 46.  56.]
[ 46.  55.]
[ 47.  52.]
```

```
[ 47.  59.]
[ 48.  51.]
[ 48.  59.]
[ 48.  50.]
[ 48.  48.]
[ 48.  59.]
[ 48.  47.]
[ 49.  55.]
[ 49.  42.]
[ 50.  49.]
[ 50.  56.]
[ 54.  47.]
[ 54.  54.]
[ 54.  53.]
[ 54.  48.]
[ 54.  52.]
[ 54.  42.]
[ 54.  51.]
[ 54.  55.]
[ 54.  41.]
[ 54.  44.]
[ 54.  57.]
[ 54.  46.]
[ 57.  58.]
[ 57.  55.]
[ 58.  60.]
[ 58.  46.]
[ 59.  55.]
[ 59.  41.]
[ 60.  49.]
[ 60.  40.]
[ 60.  42.]
[ 60.  52.]
[ 60.  47.]
[ 60.  50.]
[ 61.  42.]
[ 61.  49.]
[ 62.  41.]
[ 62.  48.]
[ 62.  59.]
[ 62.  55.]
[ 62.  56.]
[ 62.  42.]
[ 63.  50.]
[ 63.  46.]
[ 63.  43.]
[ 63.  48.]
[ 63.  52.]
```

```
[ 63.  54.]
[ 64.  42.]
[ 64.  46.]
[ 65.  48.]
[ 65.  50.]
[ 65.  43.]
[ 65.  59.]
[ 67.  43.]
[ 67.  57.]
[ 67.  56.]
[ 67.  40.]
[ 69.  58.]
[ 69.  91.]
[ 70.  29.]
[ 70.  77.]
[ 71.  35.]
[ 71.  95.]
[ 71.  11.]
[ 71.  75.]
[ 71.   9.]
[ 71.  75.]
[ 72.  34.]
[ 72.  71.]
[ 73.   5.]
[ 73.  88.]
[ 73.   7.]
[ 73.  73.]
[ 74.  10.]
[ 74.  72.]
[ 75.   5.]
[ 75.  93.]
[ 76.  40.]
[ 76.  87.]
[ 77.  12.]
[ 77.  97.]
[ 77.  36.]
[ 77.  74.]
[ 78.  22.]
[ 78.  90.]
[ 78.  17.]
[ 78.  88.]
[ 78.  20.]
[ 78.  76.]
[ 78.  16.]
[ 78.  89.]
[ 78.   1.]
[ 78.  78.]
[ 78.   1.]
```

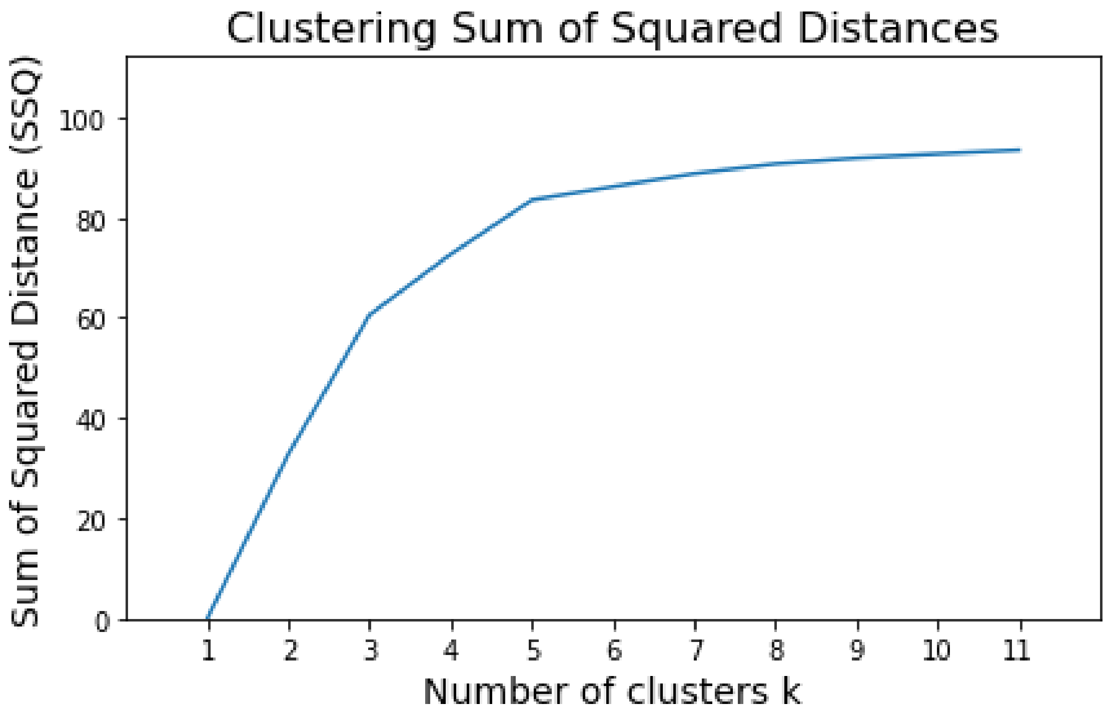
```
 [ 78.  73.]
 [ 79.  35.]
 [ 79.  83.]
 [ 81.   5.]
 [ 81.  93.]
 [ 85.  26.]
 [ 85.  75.]
 [ 86.  20.]
 [ 86.  95.]
 [ 87.  27.]
 [ 87.  63.]
 [ 87.  13.]
 [ 87.  75.]
 [ 87.  10.]
 [ 87.  92.]
 [ 88.  13.]
 [ 88.  86.]
 [ 88.  15.]
 [ 88.  69.]
 [ 93.  14.]
 [ 93.  90.]
 [ 97.  32.]
 [ 97.  86.]
 [ 98.  15.]
 [ 98.  88.]
 [ 99.  39.]
 [ 99.  97.]
 [101.  24.]
 [101.  68.]
 [103.  17.]
 [103.  85.]
 [103.  23.]
 [103.  69.]
 [113.   8.]
 [113.  91.]
 [120.  16.]
 [120.  79.]
 [126.  28.]
 [126.  74.]
 [137.  18.]
 [137.  83.]]
```

1.1 The SSQs computed for k values between 1 and 10

[3]:
```
ssqs = ssq_statistics(data, ks=range(1, 11+1))
plot_ssq_statistics(ssqs)
```

```
<ipython-input-1-eb5cd7812fff>:218: DeprecationWarning: scipy.zeros is
```

```
deprecated and will be removed in SciPy 2.0.0, use numpy.zeros instead
  ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```

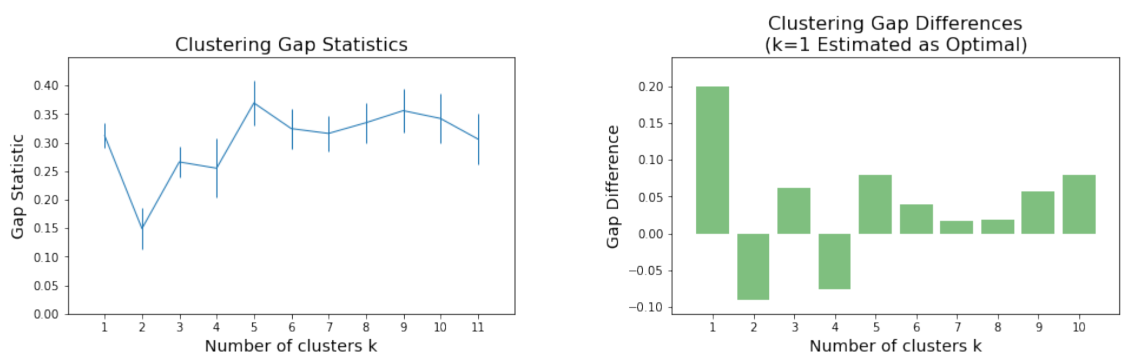


1.2 The gap statistics computed for k values between 1 and 10

[4]:
```
gaps, errs, difs = gap_statistics(data, nrefs=20, ks=range(1, 11+1))
plot_gap_statistics(gaps, errs, difs)
```

```
<ipython-input-1-eb5cd7812fff>:71: DeprecationWarning: scipy.diag is deprecated
and will be removed in SciPy 2.0.0, use numpy.diag instead
  dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input dataset
<ipython-input-1-eb5cd7812fff>:82: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
<ipython-input-1-eb5cd7812fff>:83: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
<ipython-input-1-eb5cd7812fff>:84: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length
ks-1)
<ipython-input-1-eb5cd7812fff>:98: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  refdisps = sp.zeros((rands.shape[2],))
```

```
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:114: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  for refdisp in refdisps)/float(nrefs)) * sp.sqrt(1+1/nrefs)
/Users/angie/opt/anaconda3/lib/python3.8/site-packages/scipy/cluster/vq.py:574:
UserWarning: One of the clusters is empty. Re-run kmeans with a different
initialization.
  warnings.warn("One of the clusters is empty. "
<ipython-input-1-eb5cd7812fff>:117: DeprecationWarning: scipy.array is
deprecated and will be removed in SciPy 2.0.0, use numpy.array instead
  difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in range(len(gaps)-1)])
```



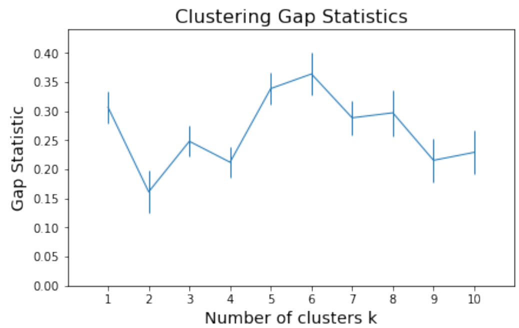
1.3 Run both measures 10 times

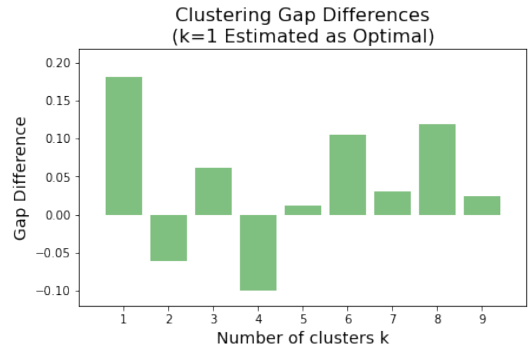
[5]:
```python
for i in range(10):
    n_clusters = 10

    gaps, errs, difs = gap_statistics(data, nrefs=20, ks=range(1,n_clusters+1))
    plot_gap_statistics(gaps, errs, difs)
```

```
    ssqs = ssq_statistics(data, ks=range(1,n_clusters+1))
    plot_ssq_statistics(ssqs)

    # Find best k
    max_gap = None
    if len(np.where(difs > 0)[0]) > 0:
        max_gap = np.where(difs > 0)[0][0] + 1 # the k with the first positive␣
↪dif

    # Fit the model on the data
    if max_gap:
        kmeans = KMeans(n_clusters=max_gap, random_state=0).fit(data)
        labels = kmeans.predict(data)
```

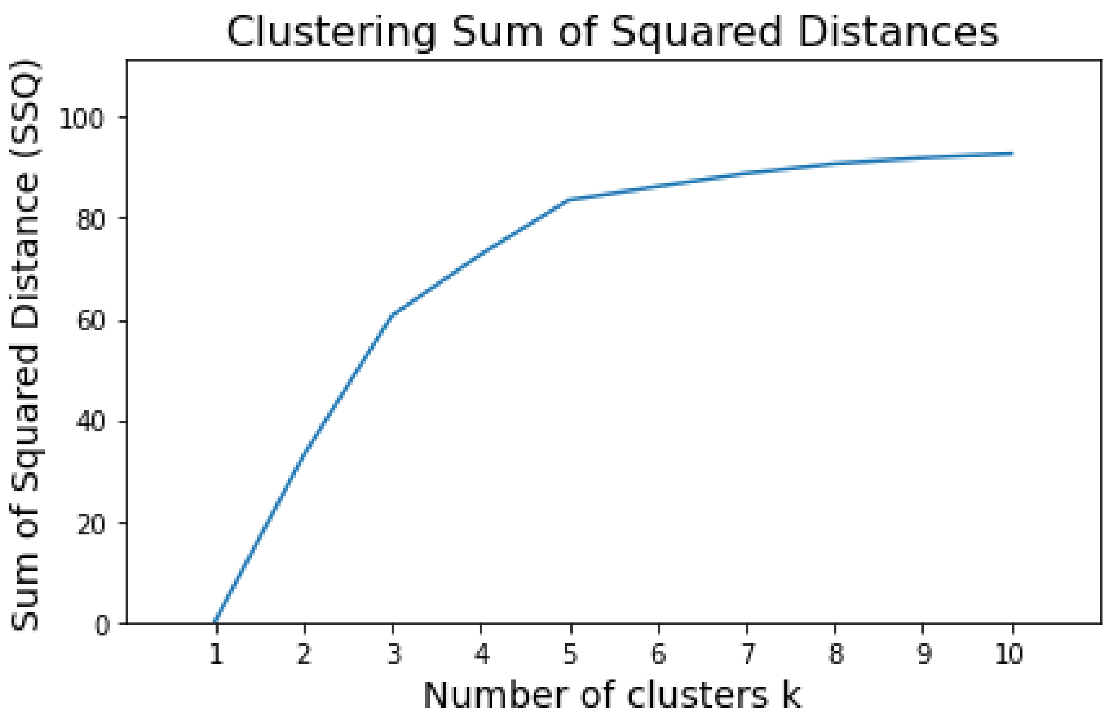
```
<ipython-input-1-eb5cd7812fff>:71: DeprecationWarning: scipy.diag is deprecated
and will be removed in SciPy 2.0.0, use numpy.diag instead
  dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input dataset
<ipython-input-1-eb5cd7812fff>:82: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
<ipython-input-1-eb5cd7812fff>:83: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
<ipython-input-1-eb5cd7812fff>:84: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length
ks-1)
<ipython-input-1-eb5cd7812fff>:98: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  refdisps = sp.zeros((rands.shape[2],))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:114: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
```

```
  for refdisp in refdisps)/float(nrefs)) * sp.sqrt(1+1/nrefs)
<ipython-input-1-eb5cd7812fff>:117: DeprecationWarning: scipy.array is
deprecated and will be removed in SciPy 2.0.0, use numpy.array instead
  difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in range(len(gaps)-1)])
```



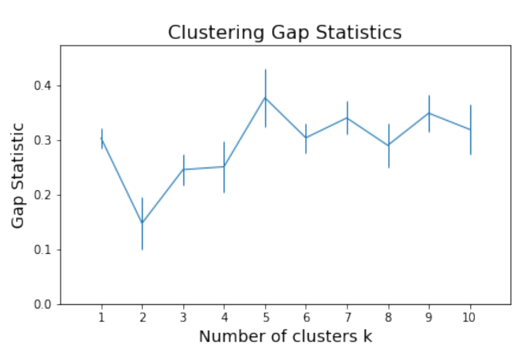


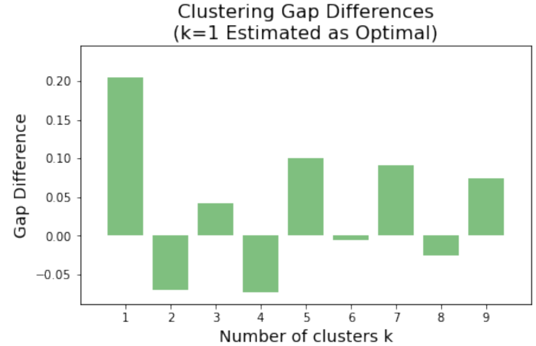

```
<ipython-input-1-eb5cd7812fff>:218: DeprecationWarning: scipy.zeros is
deprecated and will be removed in SciPy 2.0.0, use numpy.zeros instead
  ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```



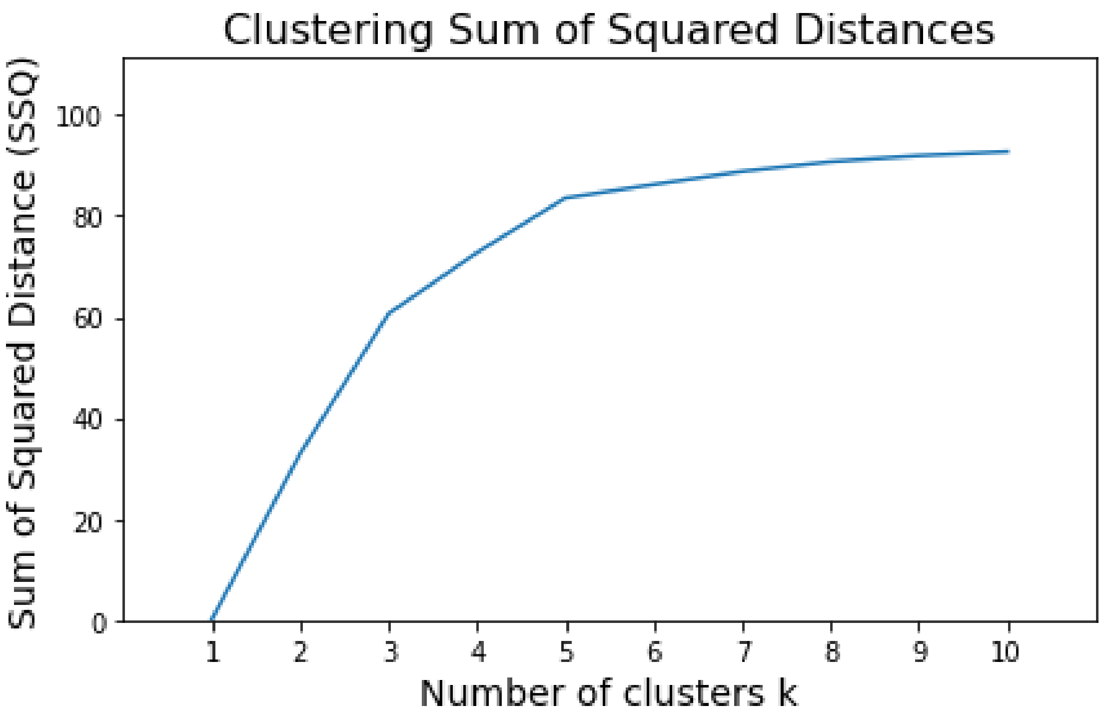

```
<ipython-input-1-eb5cd7812fff>:71: DeprecationWarning: scipy.diag is deprecated
and will be removed in SciPy 2.0.0, use numpy.diag instead
  dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input dataset
```

```
<ipython-input-1-eb5cd7812fff>:82: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
<ipython-input-1-eb5cd7812fff>:83: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
<ipython-input-1-eb5cd7812fff>:84: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length
ks-1)
<ipython-input-1-eb5cd7812fff>:98: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  refdisps = sp.zeros((rands.shape[2],))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:114: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  for refdisp in refdisps)/float(nrefs)) * sp.sqrt(1+1/nrefs)
/Users/angie/opt/anaconda3/lib/python3.8/site-packages/scipy/cluster/vq.py:574:
UserWarning: One of the clusters is empty. Re-run kmeans with a different
initialization.
  warnings.warn("One of the clusters is empty. "
<ipython-input-1-eb5cd7812fff>:117: DeprecationWarning: scipy.array is
deprecated and will be removed in SciPy 2.0.0, use numpy.array instead
  difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in range(len(gaps)-1)])
```



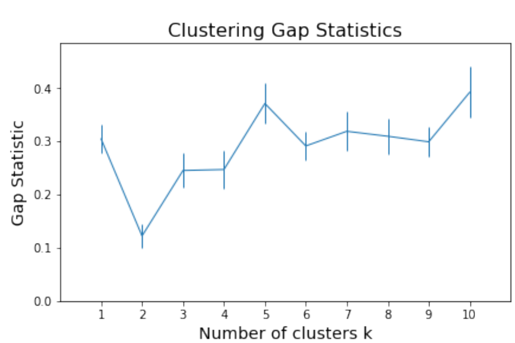


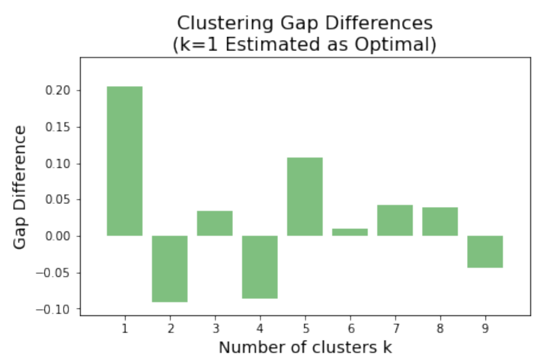

```
<ipython-input-1-eb5cd7812fff>:218: DeprecationWarning: scipy.zeros is
deprecated and will be removed in SciPy 2.0.0, use numpy.zeros instead
  ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```



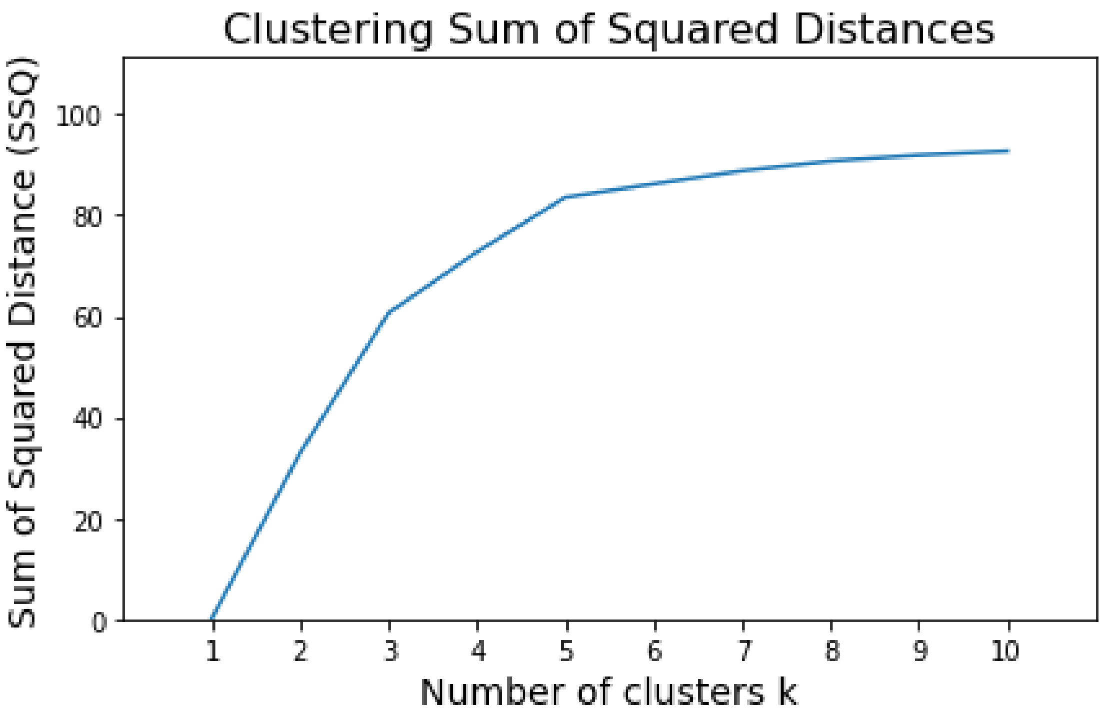

```
<ipython-input-1-eb5cd7812fff>:71: DeprecationWarning: scipy.diag is deprecated
and will be removed in SciPy 2.0.0, use numpy.diag instead
  dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input dataset
<ipython-input-1-eb5cd7812fff>:82: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
<ipython-input-1-eb5cd7812fff>:83: DeprecationWarning: scipy.zeros is deprecated
```

```
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
<ipython-input-1-eb5cd7812fff>:84: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length
ks-1)
<ipython-input-1-eb5cd7812fff>:98: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  refdisps = sp.zeros((rands.shape[2],))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:114: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  for refdisp in refdisps)/float(nrefs)) * sp.sqrt(1+1/nrefs)
/Users/angie/opt/anaconda3/lib/python3.8/site-packages/scipy/cluster/vq.py:574:
UserWarning: One of the clusters is empty. Re-run kmeans with a different
initialization.
  warnings.warn("One of the clusters is empty. "
<ipython-input-1-eb5cd7812fff>:117: DeprecationWarning: scipy.array is
deprecated and will be removed in SciPy 2.0.0, use numpy.array instead
  difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in range(len(gaps)-1)])
```



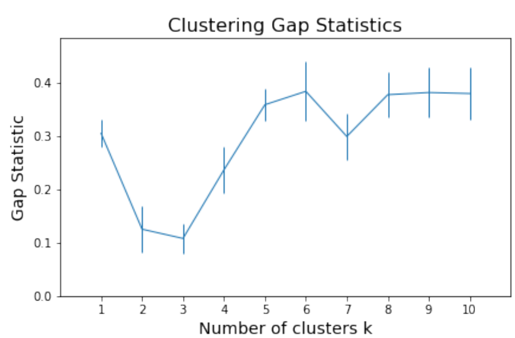


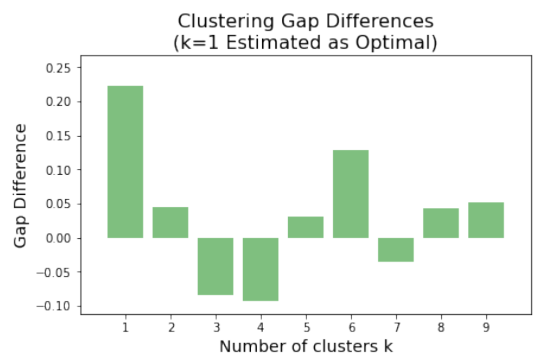
```
<ipython-input-1-eb5cd7812fff>:218: DeprecationWarning: scipy.zeros is
deprecated and will be removed in SciPy 2.0.0, use numpy.zeros instead
  ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```



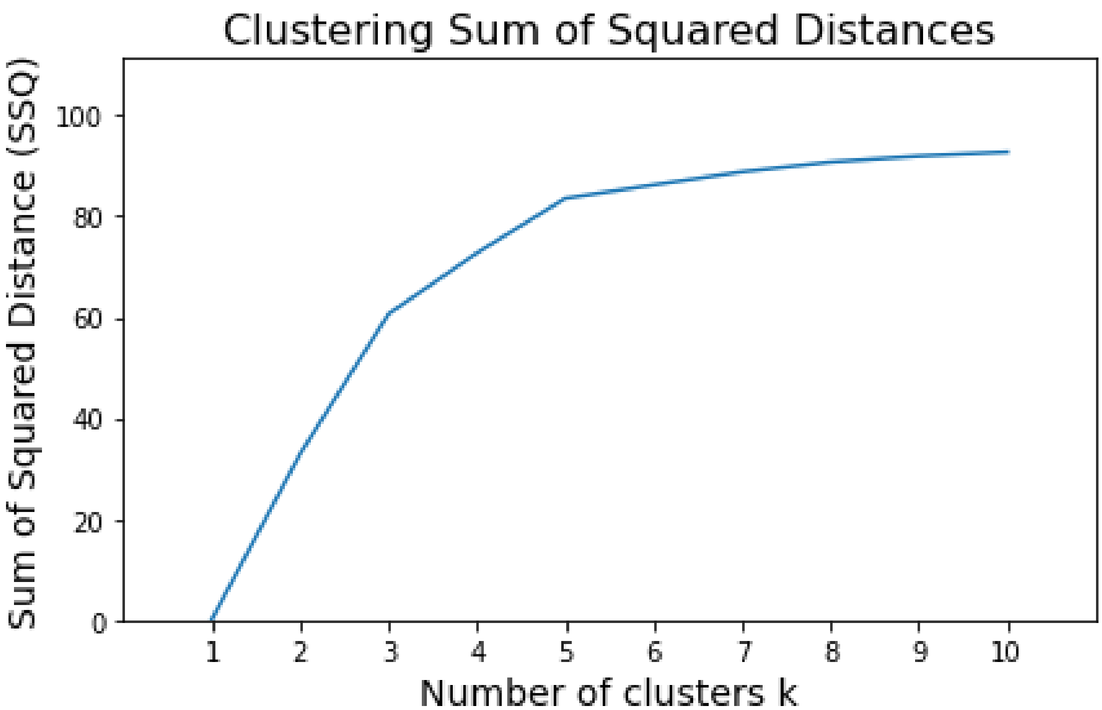

```
<ipython-input-1-eb5cd7812fff>:71: DeprecationWarning: scipy.diag is deprecated
and will be removed in SciPy 2.0.0, use numpy.diag instead
  dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input dataset
<ipython-input-1-eb5cd7812fff>:82: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
<ipython-input-1-eb5cd7812fff>:83: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
<ipython-input-1-eb5cd7812fff>:84: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length
ks-1)
<ipython-input-1-eb5cd7812fff>:98: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  refdisps = sp.zeros((rands.shape[2],))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.mean is deprecated
```

```
and will be removed in SciPy 2.0.0, use numpy.mean instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:114: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  for refdisp in refdisps)/float(nrefs)) * sp.sqrt(1+1/nrefs)
/Users/angie/opt/anaconda3/lib/python3.8/site-packages/scipy/cluster/vq.py:574:
UserWarning: One of the clusters is empty. Re-run kmeans with a different
initialization.
  warnings.warn("One of the clusters is empty. "
<ipython-input-1-eb5cd7812fff>:117: DeprecationWarning: scipy.array is
deprecated and will be removed in SciPy 2.0.0, use numpy.array instead
  difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in range(len(gaps)-1)])
```



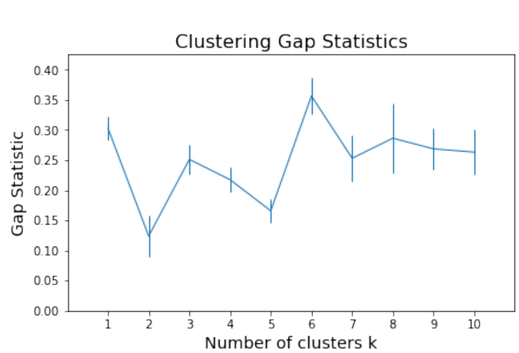


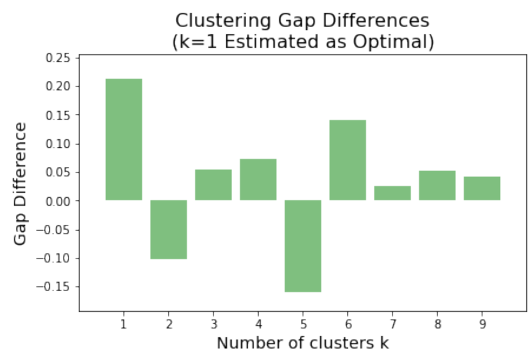

```
<ipython-input-1-eb5cd7812fff>:218: DeprecationWarning: scipy.zeros is
deprecated and will be removed in SciPy 2.0.0, use numpy.zeros instead
  ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```



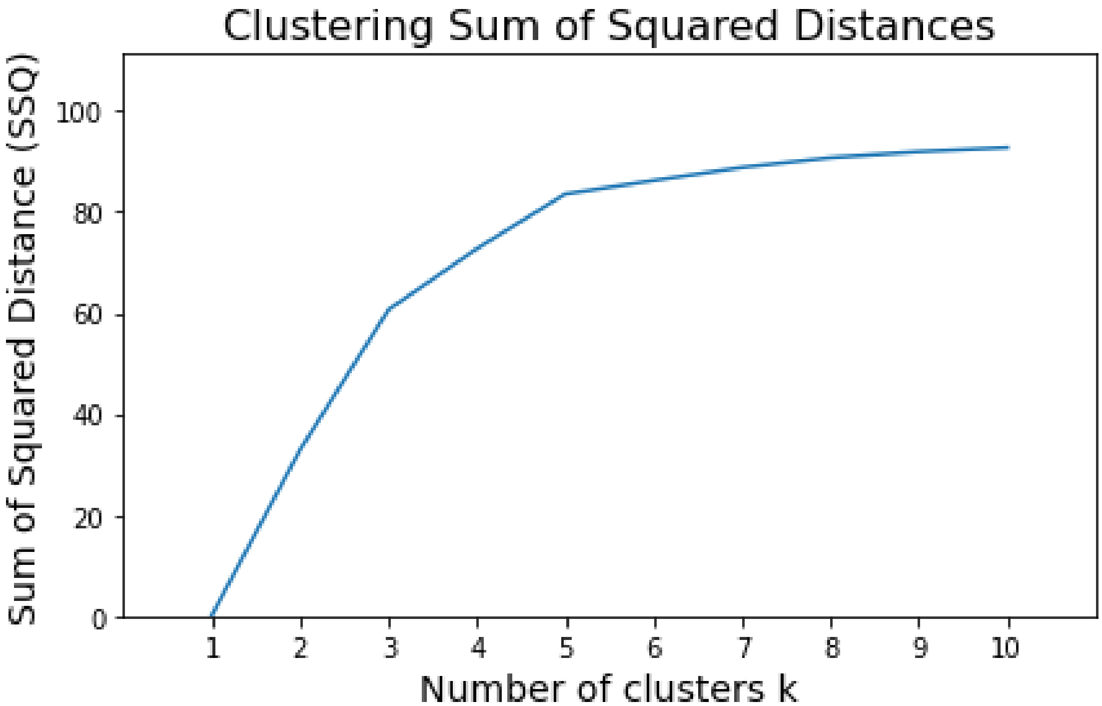

```
<ipython-input-1-eb5cd7812fff>:71: DeprecationWarning: scipy.diag is deprecated
and will be removed in SciPy 2.0.0, use numpy.diag instead
  dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input dataset
<ipython-input-1-eb5cd7812fff>:82: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
<ipython-input-1-eb5cd7812fff>:83: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
<ipython-input-1-eb5cd7812fff>:84: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length
ks-1)
<ipython-input-1-eb5cd7812fff>:98: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  refdisps = sp.zeros((rands.shape[2],))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
```

```
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:114: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  for refdisp in refdisps)/float(nrefs)) * sp.sqrt(1+1/nrefs)
/Users/angie/opt/anaconda3/lib/python3.8/site-packages/scipy/cluster/vq.py:574:
UserWarning: One of the clusters is empty. Re-run kmeans with a different
initialization.
  warnings.warn("One of the clusters is empty. "
<ipython-input-1-eb5cd7812fff>:117: DeprecationWarning: scipy.array is
deprecated and will be removed in SciPy 2.0.0, use numpy.array instead
  difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in range(len(gaps)-1)])
```



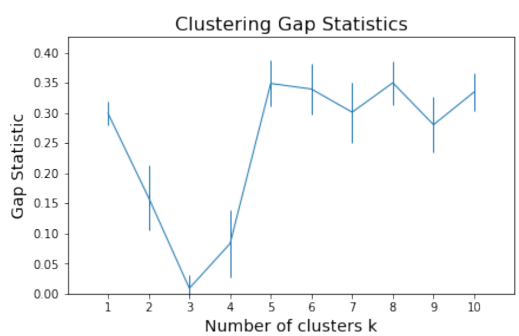


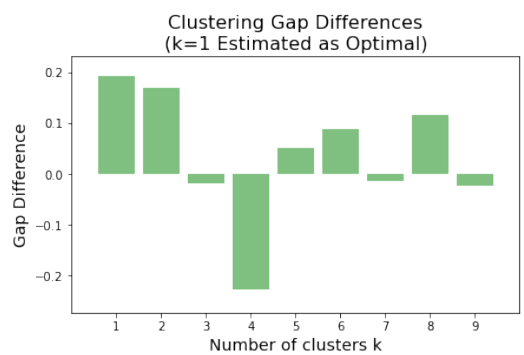

```
<ipython-input-1-eb5cd7812fff>:218: DeprecationWarning: scipy.zeros is
deprecated and will be removed in SciPy 2.0.0, use numpy.zeros instead
  ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```



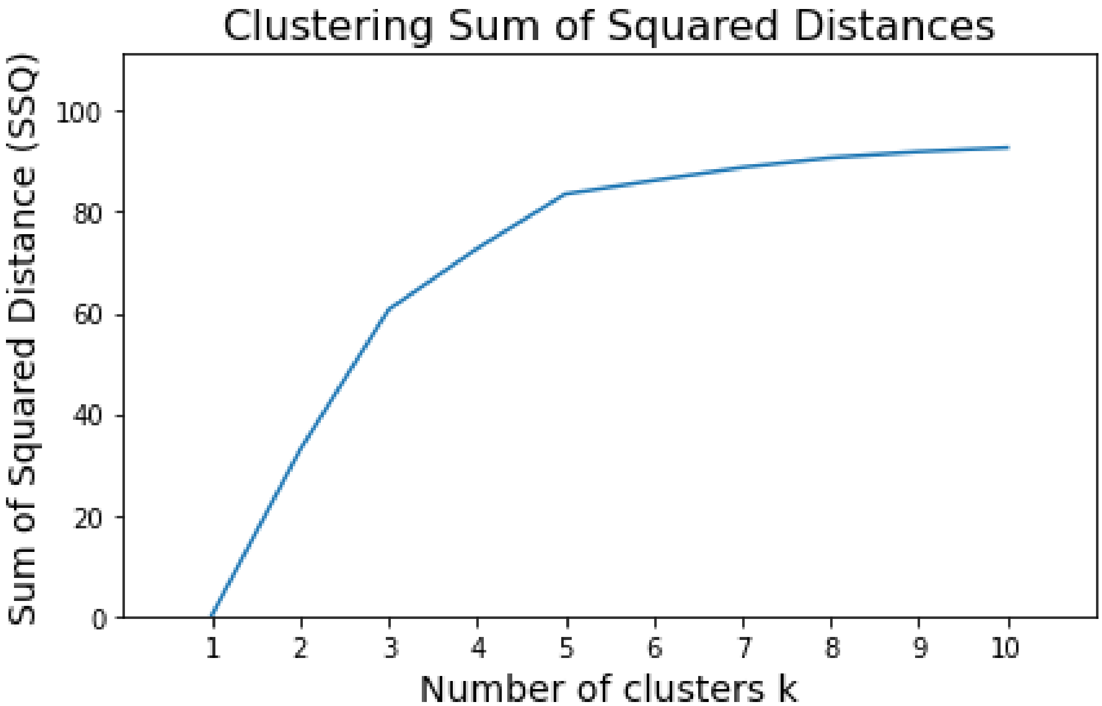

```
<ipython-input-1-eb5cd7812fff>:71: DeprecationWarning: scipy.diag is deprecated
and will be removed in SciPy 2.0.0, use numpy.diag instead
  dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input dataset
<ipython-input-1-eb5cd7812fff>:82: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
<ipython-input-1-eb5cd7812fff>:83: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
<ipython-input-1-eb5cd7812fff>:84: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length
ks-1)
<ipython-input-1-eb5cd7812fff>:98: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  refdisps = sp.zeros((rands.shape[2],))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
```

```
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:114: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  for refdisp in refdisps)/float(nrefs)) * sp.sqrt(1+1/nrefs)
/Users/angie/opt/anaconda3/lib/python3.8/site-packages/scipy/cluster/vq.py:574:
UserWarning: One of the clusters is empty. Re-run kmeans with a different
initialization.
  warnings.warn("One of the clusters is empty. "
<ipython-input-1-eb5cd7812fff>:117: DeprecationWarning: scipy.array is
deprecated and will be removed in SciPy 2.0.0, use numpy.array instead
  difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in range(len(gaps)-1)])
```



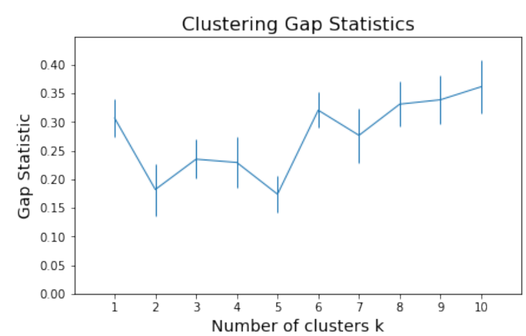


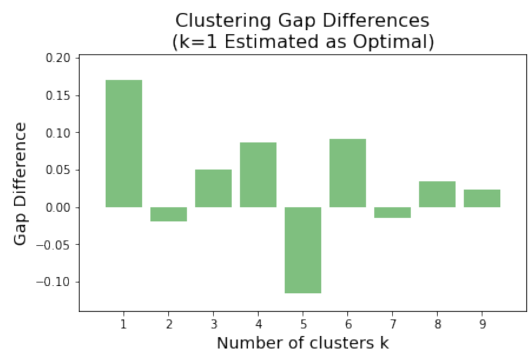

```
<ipython-input-1-eb5cd7812fff>:218: DeprecationWarning: scipy.zeros is
deprecated and will be removed in SciPy 2.0.0, use numpy.zeros instead
  ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```



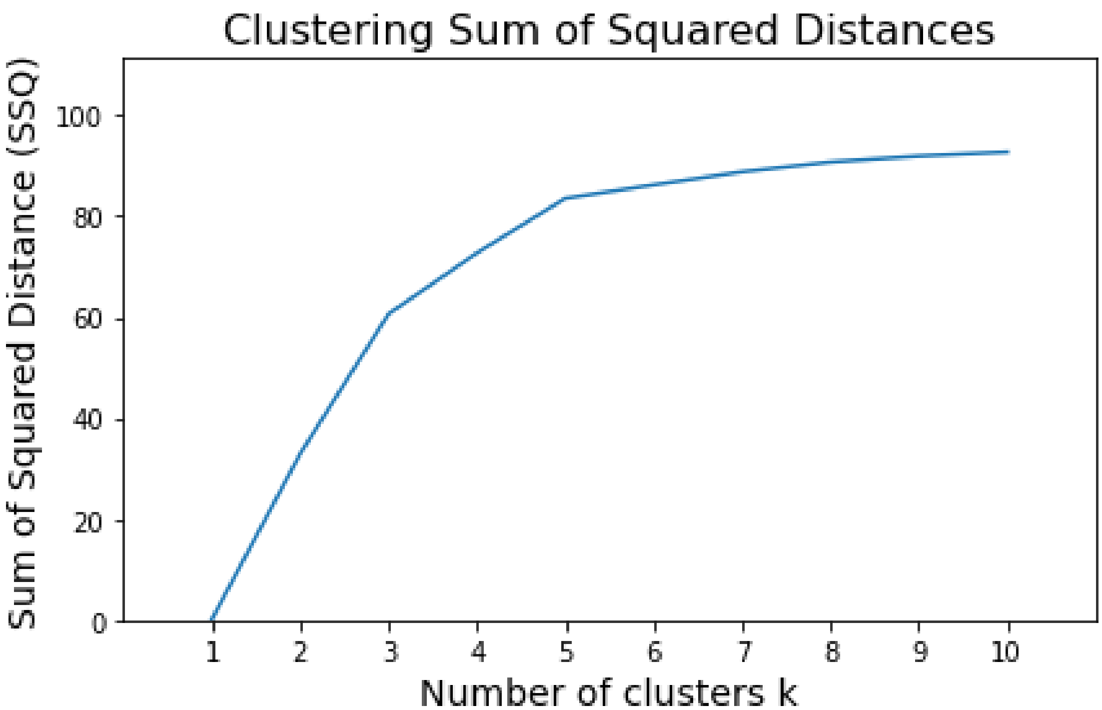

```
<ipython-input-1-eb5cd7812fff>:71: DeprecationWarning: scipy.diag is deprecated
and will be removed in SciPy 2.0.0, use numpy.diag instead
  dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input dataset
<ipython-input-1-eb5cd7812fff>:82: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
<ipython-input-1-eb5cd7812fff>:83: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
<ipython-input-1-eb5cd7812fff>:84: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length
ks-1)
<ipython-input-1-eb5cd7812fff>:98: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  refdisps = sp.zeros((rands.shape[2],))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
```

```
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:114: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  for refdisp in refdisps)/float(nrefs)) * sp.sqrt(1+1/nrefs)
/Users/angie/opt/anaconda3/lib/python3.8/site-packages/scipy/cluster/vq.py:574:
UserWarning: One of the clusters is empty. Re-run kmeans with a different
initialization.
  warnings.warn("One of the clusters is empty. "
<ipython-input-1-eb5cd7812fff>:117: DeprecationWarning: scipy.array is
deprecated and will be removed in SciPy 2.0.0, use numpy.array instead
  difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in range(len(gaps)-1)])
```



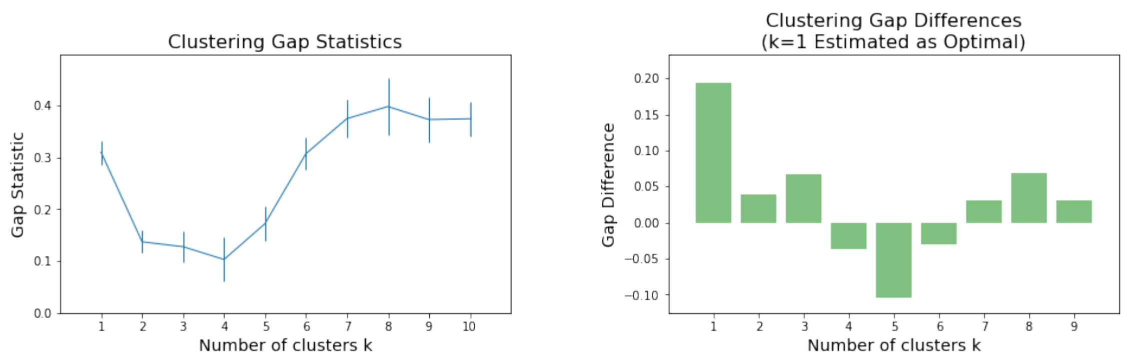



```
<ipython-input-1-eb5cd7812fff>:218: DeprecationWarning: scipy.zeros is
deprecated and will be removed in SciPy 2.0.0, use numpy.zeros instead
  ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```

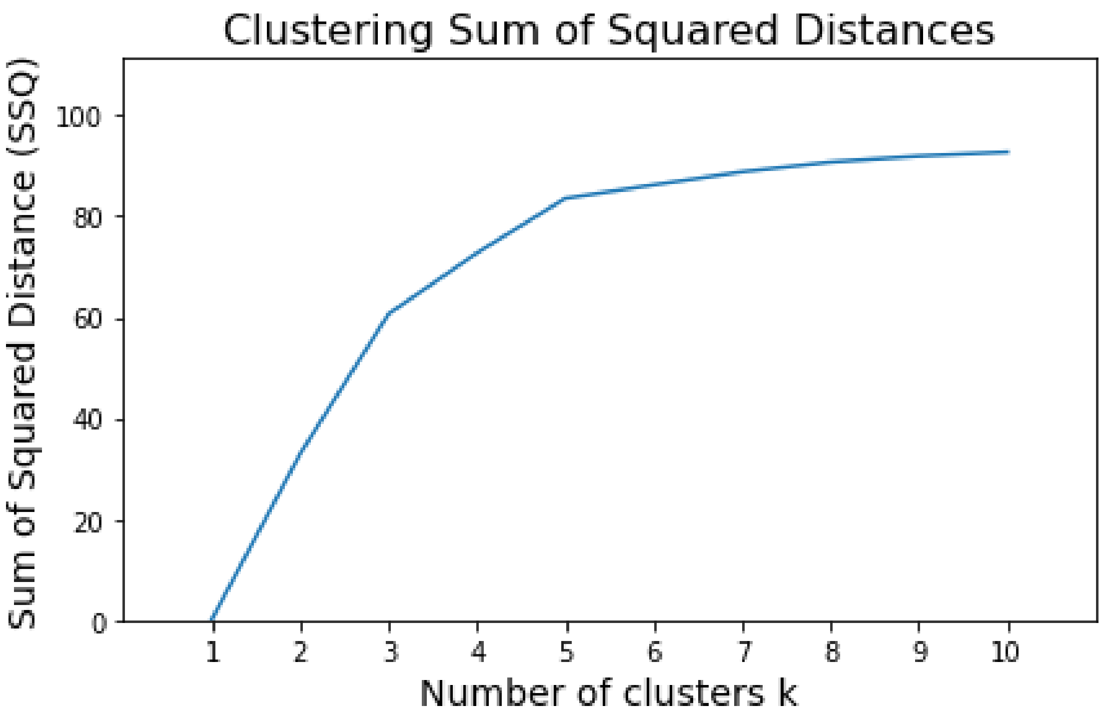



```
<ipython-input-1-eb5cd7812fff>:71: DeprecationWarning: scipy.diag is deprecated
and will be removed in SciPy 2.0.0, use numpy.diag instead
  dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input dataset
<ipython-input-1-eb5cd7812fff>:82: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
<ipython-input-1-eb5cd7812fff>:83: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
<ipython-input-1-eb5cd7812fff>:84: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length
ks-1)
<ipython-input-1-eb5cd7812fff>:98: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  refdisps = sp.zeros((rands.shape[2],))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
```

```
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:114: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  for refdisp in refdisps)/float(nrefs)) * sp.sqrt(1+1/nrefs)
/Users/angie/opt/anaconda3/lib/python3.8/site-packages/scipy/cluster/vq.py:574:
UserWarning: One of the clusters is empty. Re-run kmeans with a different
initialization.
  warnings.warn("One of the clusters is empty. "
<ipython-input-1-eb5cd7812fff>:117: DeprecationWarning: scipy.array is
deprecated and will be removed in SciPy 2.0.0, use numpy.array instead
  difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in range(len(gaps)-1)])
```

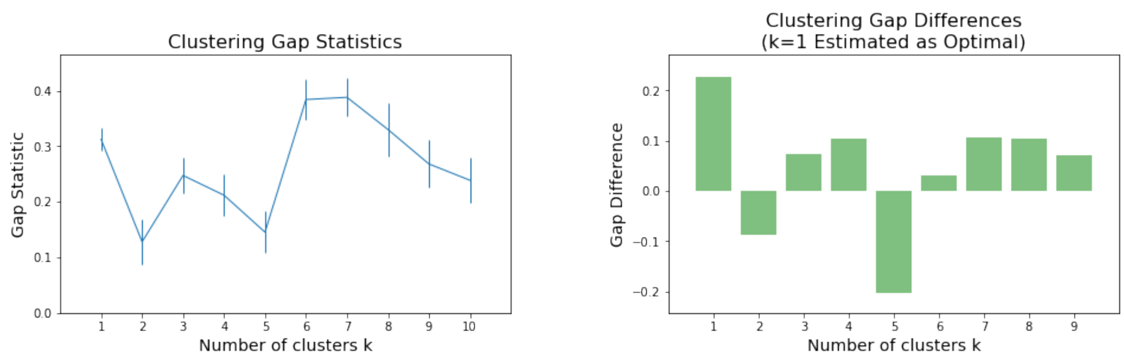



```
<ipython-input-1-eb5cd7812fff>:218: DeprecationWarning: scipy.zeros is
deprecated and will be removed in SciPy 2.0.0, use numpy.zeros instead
  ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```

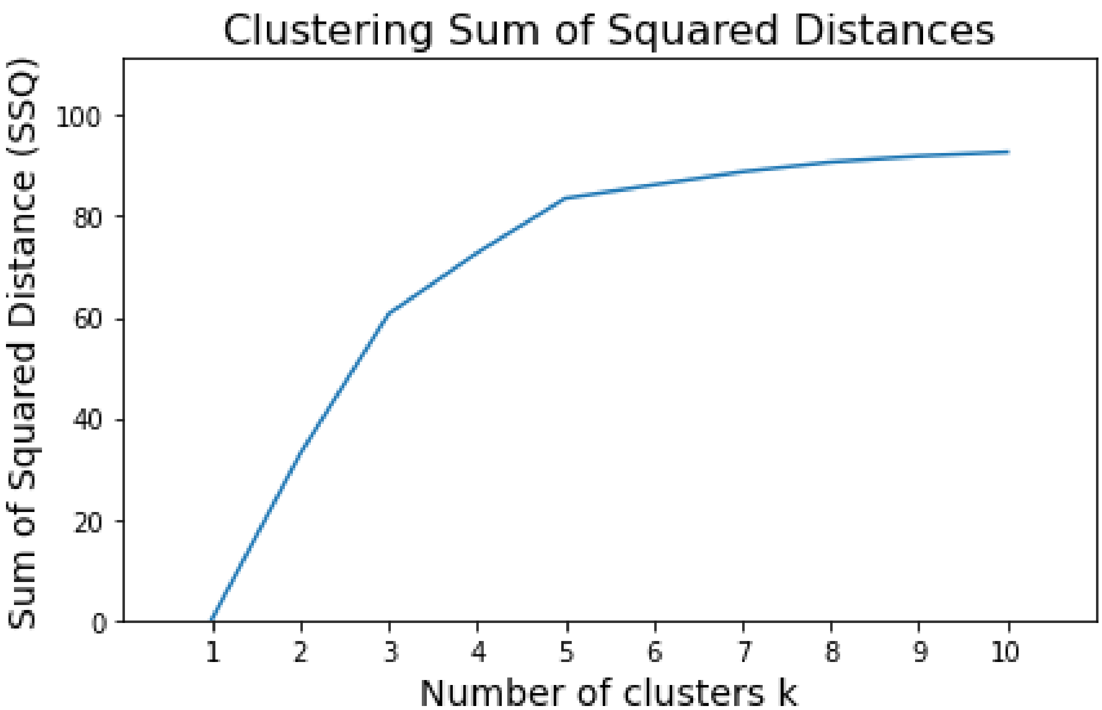



```
<ipython-input-1-eb5cd7812fff>:71: DeprecationWarning: scipy.diag is deprecated
and will be removed in SciPy 2.0.0, use numpy.diag instead
  dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input dataset
<ipython-input-1-eb5cd7812fff>:82: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
<ipython-input-1-eb5cd7812fff>:83: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
<ipython-input-1-eb5cd7812fff>:84: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length
ks-1)
<ipython-input-1-eb5cd7812fff>:98: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  refdisps = sp.zeros((rands.shape[2],))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
```

```
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:114: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  for refdisp in refdisps)/float(nrefs)) * sp.sqrt(1+1/nrefs)
/Users/angie/opt/anaconda3/lib/python3.8/site-packages/scipy/cluster/vq.py:574:
UserWarning: One of the clusters is empty. Re-run kmeans with a different
initialization.
  warnings.warn("One of the clusters is empty. "
<ipython-input-1-eb5cd7812fff>:117: DeprecationWarning: scipy.array is
deprecated and will be removed in SciPy 2.0.0, use numpy.array instead
  difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in range(len(gaps)-1)])
```

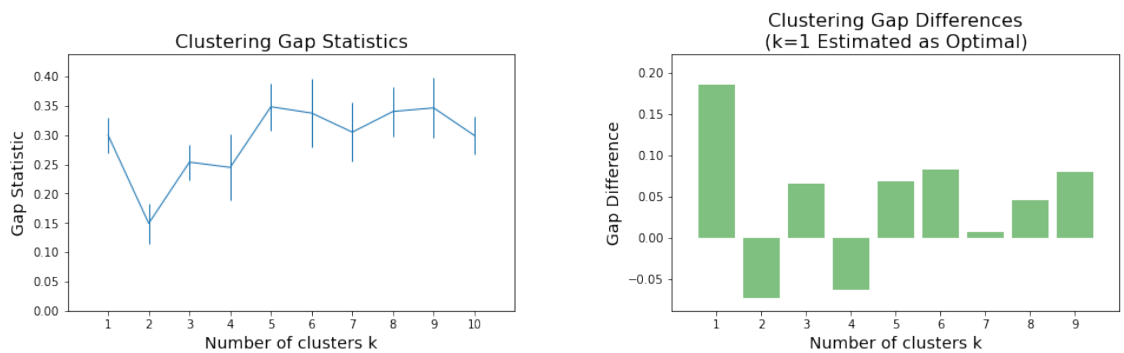



```
<ipython-input-1-eb5cd7812fff>:218: DeprecationWarning: scipy.zeros is
deprecated and will be removed in SciPy 2.0.0, use numpy.zeros instead
  ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```

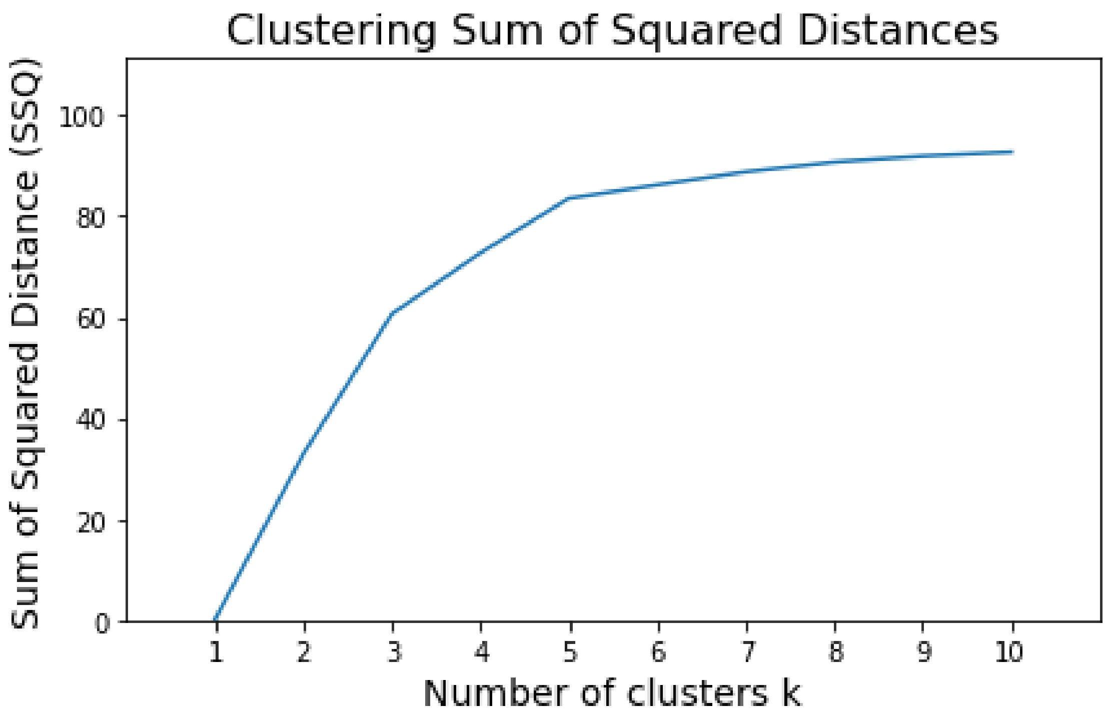



```
<ipython-input-1-eb5cd7812fff>:71: DeprecationWarning: scipy.diag is deprecated
and will be removed in SciPy 2.0.0, use numpy.diag instead
  dists = sp.matrix(sp.diag(tops-bots)) # the bounding box of the input dataset
<ipython-input-1-eb5cd7812fff>:82: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  gaps = sp.zeros((len(ks),))   # array for gap statistics (lenth ks)
<ipython-input-1-eb5cd7812fff>:83: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  errs = sp.zeros((len(ks),))   # array for model standard errors (length ks)
<ipython-input-1-eb5cd7812fff>:84: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  difs = sp.zeros((len(ks)-1,)) # array for differences between gaps (length
ks-1)
<ipython-input-1-eb5cd7812fff>:98: DeprecationWarning: scipy.zeros is deprecated
and will be removed in SciPy 2.0.0, use numpy.zeros instead
  refdisps = sp.zeros((rands.shape[2],))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:110: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  gaps[i] = sp.mean(sp.log(refdisps) - sp.log(disp))
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.log is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.log instead
```

```
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.mean is deprecated
and will be removed in SciPy 2.0.0, use numpy.mean instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:113: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  errs[i] = sp.sqrt(sum(((sp.log(refdisp)-sp.mean(sp.log(refdisps)))**2) \
<ipython-input-1-eb5cd7812fff>:114: DeprecationWarning: scipy.sqrt is deprecated
and will be removed in SciPy 2.0.0, use numpy.lib.scimath.sqrt instead
  for refdisp in refdisps)/float(nrefs)) * sp.sqrt(1+1/nrefs)
/Users/angie/opt/anaconda3/lib/python3.8/site-packages/scipy/cluster/vq.py:574:
UserWarning: One of the clusters is empty. Re-run kmeans with a different
initialization.
  warnings.warn("One of the clusters is empty. "
<ipython-input-1-eb5cd7812fff>:117: DeprecationWarning: scipy.array is
deprecated and will be removed in SciPy 2.0.0, use numpy.array instead
  difs = sp.array([gaps[k] - (gaps[k+1]-errs[k+1]) for k in range(len(gaps)-1)])
```



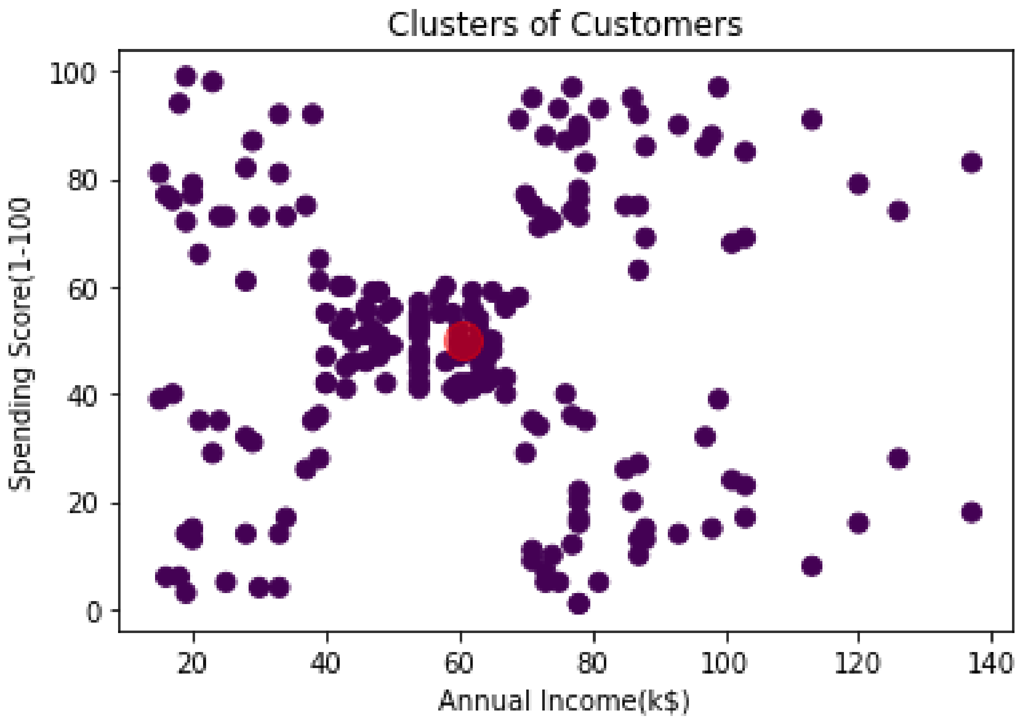

```
<ipython-input-1-eb5cd7812fff>:218: DeprecationWarning: scipy.zeros is
deprecated and will be removed in SciPy 2.0.0, use numpy.zeros instead
  ssqs = sp.zeros((len(ks),)) # array for SSQs (lenth ks)
```



1.4 Scatter plot of the data in 2d showing the clusters in different colors

[6]:
```
# k=5
kmeans1 = KMeans(n_clusters=5)
kmeans1.fit(data)
labels1 = kmeans1.predict(data)
plt.scatter(data[:, 0], data[:, 1], c=labels1, s=50, cmap='viridis')
plt.title('Clusters of Customers')
plt.xlabel('Annual Income(k$)')
plt.ylabel('Spending Score(1-100')
centers = kmeans1.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200, alpha=0.5);
```


```
[7]: # k=1
kmeans2 = KMeans(n_clusters=1)
kmeans2.fit(data)
labels2 = kmeans2.predict(data)
plt.scatter(data[:, 0], data[:, 1], c=labels2, s=50, cmap='viridis')
plt.title('Clusters of Customers')
plt.xlabel('Annual Income(k$)')
plt.ylabel('Spending Score(1-100')
centers = kmeans2.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200, alpha=0.5);
```



Q1. Where did you estimate the elbow point to be (between what values of k)? What value of k was typically estimated as optimal by the gap statistic? To adequately answer this question, consider generating both measures several (at least 5) times, as there may be some amount of variation in the value of k that they each estimate as optimal.

The SSQ elbow point is estimated to be 5. The gap statistics typically estimate k=1 as optimal.

Q2. Based on the scatter plot of the clustered data, what makes most sense? Give logical interpretation from visually inspecting the clusters.

Visually, k=5 makes more sense since on the clustered data scatter around each center while the clusters are seperate from each other.

Q3. Between SSQ and Gap Statistics, does one measure seem to be a consistently better criterion for choosing the value of k than the other?

In this case, it seems SSQ is consistently better than Gap Statistics for choosing the value of k. But in general, it's difficult to say that one criterion is consistently better than the other because sometimes it's difficult to identify the "elbow" in elbow method. And sometimes it's difficult to get a serious of reference distributions required by the gap statistics.