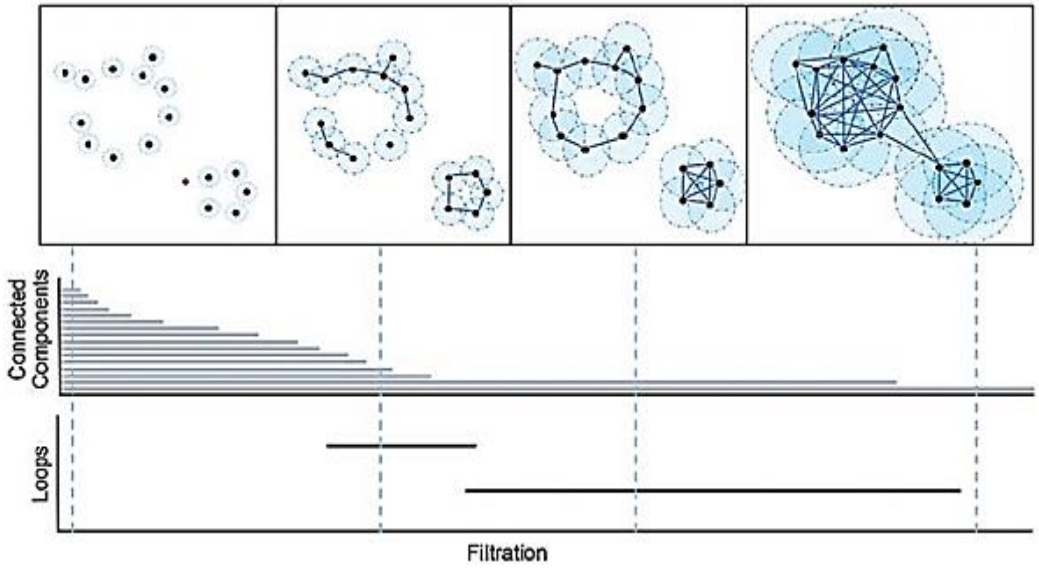
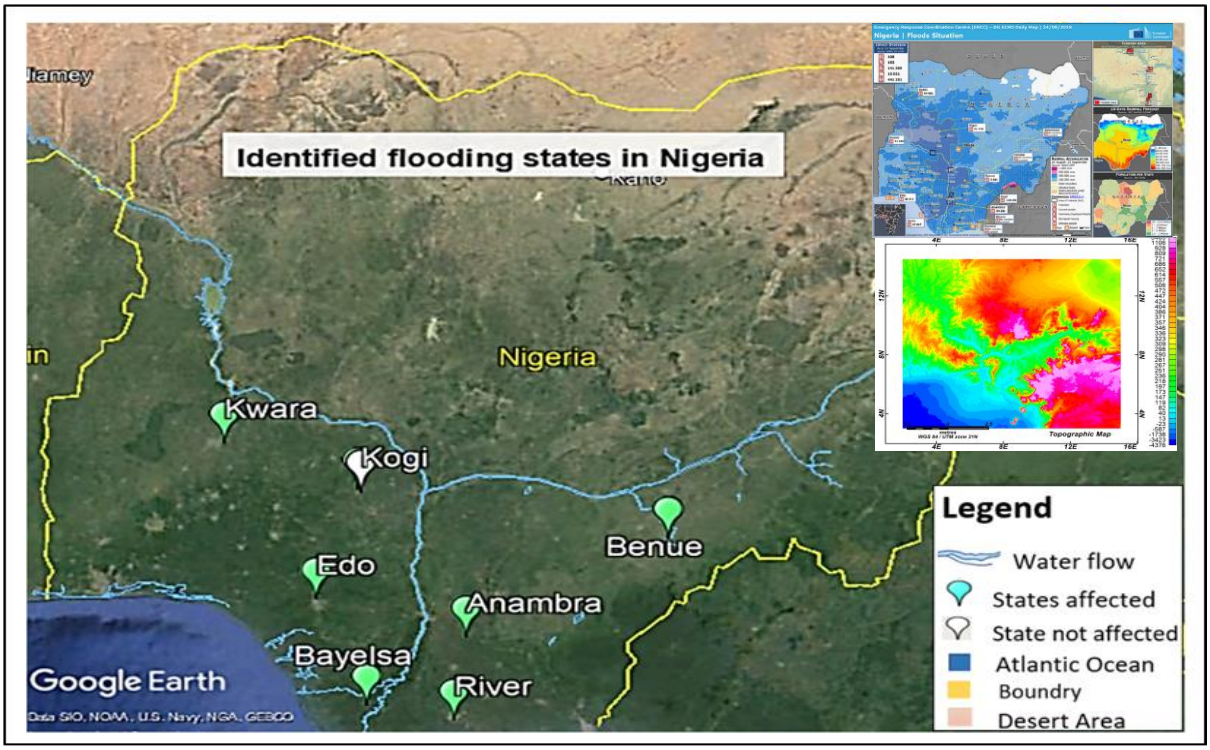
2.3 Computation of Persistence homology (PH) in the TDA-ML approach



Filteration and corresponding Barcodes

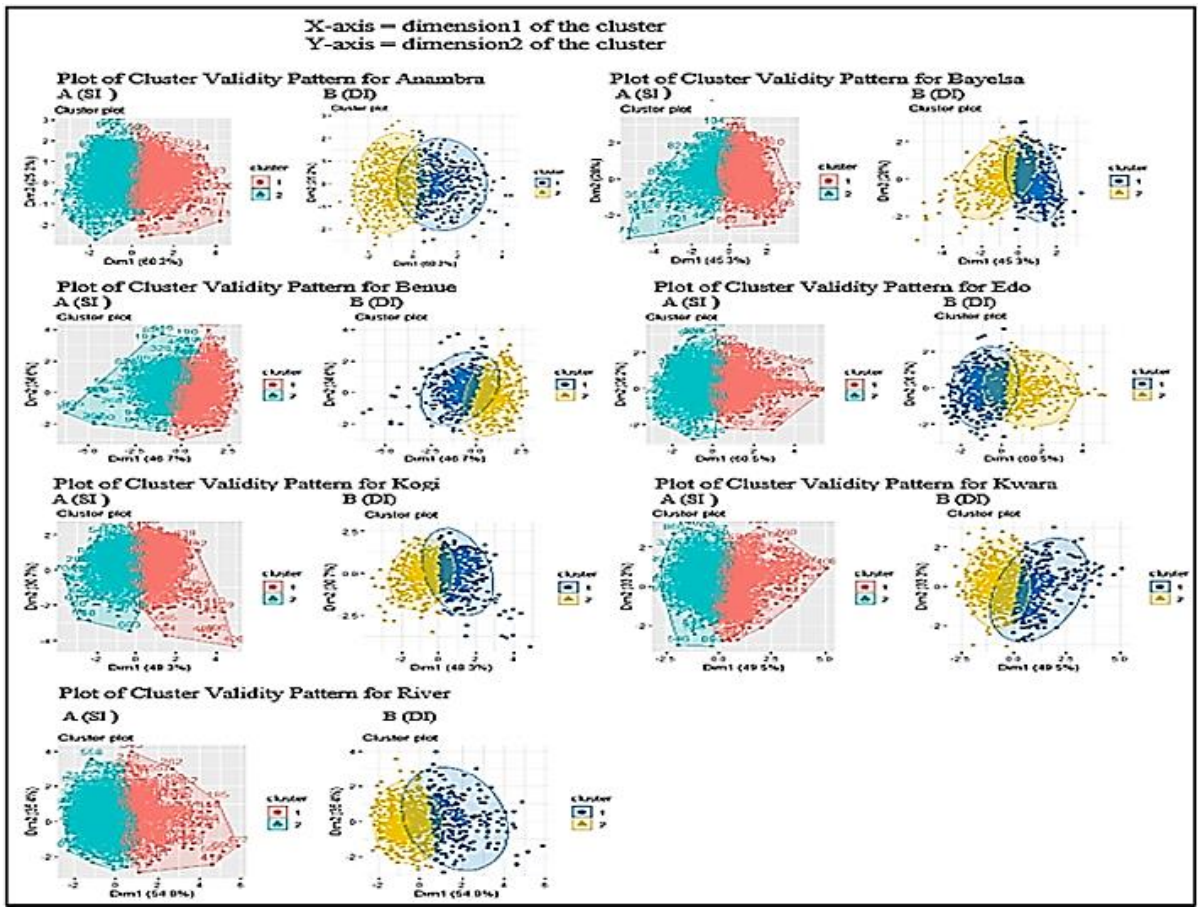
3.1. Discussions on K-mean Clustering of the Hybrid Method



Predicted Flooding States in Nigeria

Cluster Validity Tests Results for Dunn and Silouette Indices

State	Dunn index	Silhouette index
Anambra	0.8123	0.8619
Bayelsa	0.7913	0.8758
Benue	0.3391	0.3794
Edo	0.6357	0.6810
Kogi	0.5879	0.6118
Kwara	0.3529	0.3804
River	0.8112	0.9117



Plots of the Cluster, k = 2 Validity Feature Pattern for the Seven States

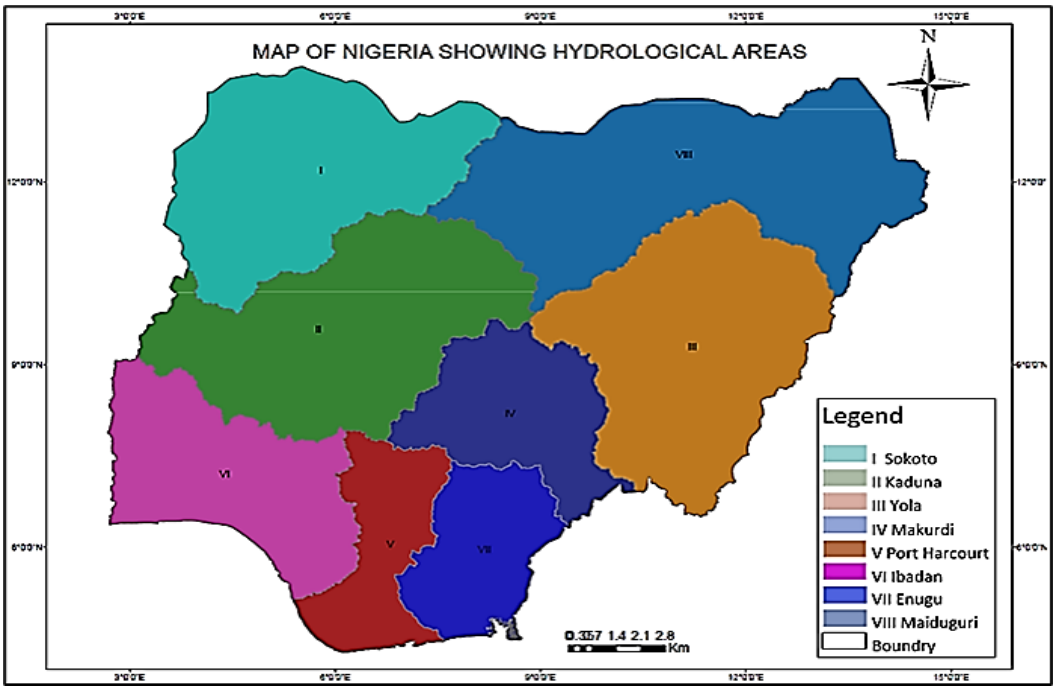
Table 5. Summary Statistics of Topological Features for Dimension 0 on September

State	Sum of all lifetimes ($\mathbf{sum_D}$)	Avg. of lifetime of all holes ($\boldsymbol{avg_D}$)
Anambra	565.0383	19.4841
Bayelsa	552.9779	23.0408
Benue	562.8185	26.8009
Edo	563.3173	25.6053
Kogi	597.2725	29.8637
Kwara	629.9208	26.2467
River	566.9522	25.7705

Table 6. Summary Statistics of Topological Features for Dimension 1 on September

State	Max. holes lifetime ($\mathbf{max_D}$)	Max. death point of holes ($\boldsymbol{max_d}$)	Sum of all lifetimes($\mathbf{sum_D}$)	Avg. of lifetime of all holes ($\boldsymbol{avg_D}$)
Anambra	0.2343	21.3577	0.5986	0.4982
Bayelsa	0.4142	1.4142	2.0710	0.4142
Benue	0.3377	19.416	0.3387	0.3377
Edo	0.4142	15.414	1.6568	0.4442
Kogi	0.3377	1.4142	1.2726	0.4150
Kwara	0.4146	3.4112	1.0913	0.4143
River	0.4244	3.6070	0.7066	0.3027

3.4. Classification of the Hydrological Areas



Map of Nigeria Showing Hydrological Areas

APPENDIX: Code for Data Presentation

```
import matplotlib.pyplot as plt
from matplotlib import style
style.use('ggplot')
import numpy as np
from sklearn.cluster import KMeans
from sklearn.cluster import MeanShift
import pandas as pd
from sklearn.preprocessing import scale
import sklearn.metrics as sm
import time
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import recall_score
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn import preprocessing

data = pd.read_csv('Anambra11_Use_CLustered.csv')# reads the csv file
names  = ['Date', 'Month Number','Monthly Maximum Temperature(K)', 'Monthly Minimum
Temperature(K)', 'Precipitation (mm)', 'Monthly Average Wind Speed (m/s)', 'Clusters']#defines
columns in the dataset

ax = plt.gca()

#data.hist()
data.plot(kind='density', subplots=True, layout=(3,3), sharex=False)
#data.plot(kind='box', subplots=True, layout=(3,3), sharex=False, sharey=False)

#plt.show()
#print((data.shape))
#print(data.Clusters.value_counts())

#plt.legend(loc='lower center', bbox_to_anchor=(0.5, 1.0),
#      ncol=3, fancybox=True, shadow=True)
# plt.grid(True)#removes grid lines from plots
ax.xaxis.grid()
plt.axis('on')
ax = plt.gca()#activates the use of ax-set_facecolor
ax.set_facecolor('white')

#code segment to enable borders
for pos in ['top', 'bottom', 'right', 'left']:
      ax.spines[pos].set_edgecolor('black')
```

Code For K-Means Clustering and Silhouette Analysis

```
import numpy as np
import pandas as pd

from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score

import matplotlib.pyplot as plt
import matplotlib.cm as cm
import seaborn as sns

print(__doc__)

input_data = pd.read_csv('Anambra11_Use.csv')

x = input_data.iloc[:,].values
#x = dataset.iloc[:,].values

for n_clusters in range(2,11):
    # Create a subplot with 1 row and 2 columns
    fig, (ax1, ax2) = plt.subplots(1, 2)
    fig.set_size_inches(18, 7)

    # The 1st subplot is the silhouette plot
    # The silhouette coefficient can range from -1, 1 but in this example all
    # lie within [-0.1, 1]
    ax1.set_xlim([-0.1, 1])
    # The (n_clusters+1)*10 is for inserting blank space between silhouette
    # plots of individual clusters, to demarcate them clearly.
    ax1.set_ylim([0, len(x) + (n_clusters + 1) * 10])

    # Initialize the clusterer with n_clusters value and a random generator
    # seed of 10 for reproducibility.
    clusterer = KMeans(n_clusters=n_clusters, random_state=10  )
    cluster_labels = clusterer.fit_predict(x)

    # The silhouette_score gives the average value for all the samples.
    # This gives a perspective into the density and separation of the formed
    # clusters
    silhouette_avg = silhouette_score(x, cluster_labels)
    print("For n_clusters =", n_clusters,
          "The average silhouette_score is :", silhouette_avg)

    # Compute the silhouette scores for each sample
    sample_silhouette_values = silhouette_samples(x, cluster_labels)

    y_lower = 10
    for i in range(n_clusters):
        # Aggregate the silhouette scores for samples belonging to
        # cluster i, and sort them
        ith_cluster_silhouette_values = \
            sample_silhouette_values[cluster_labels == i]

        ith_cluster_silhouette_values.sort()

        size_cluster_i = ith_cluster_silhouette_values.shape[0]
        y_upper = y_lower + size_cluster_i

        color = cm.nipy_spectral(float(i) / n_clusters)
```

```
        ax1.fill_betweenx(np.arange(y_lower, y_upper),
                          0, ith_cluster_silhouette_values,
                          facecolor=color, edgecolor=color, alpha=0.7)

        # Label the silhouette plots with their cluster numbers at the middle
      #  ax1.text(-0.05, y_lower + 0.5 * size_cluster_i, str(i))

        # Compute the new y_lower for next plot
        y_lower = y_upper + 10  # 10 for the 0 samples

    ax1.set_title("The silhouette plot for the various clusters.")
    ax1.set_xlabel("The silhouette coefficient values")
    ax1.set_ylabel("Cluster label")

    # The vertical line for average silhouette score of all the values
    ax1.axvline(x=silhouette_avg, color="red", linestyle="--")

    ax1.set_yticks([])  # Clear the yaxis labels / ticks
    ax1.set_xticks([-0.1, 0, 0.2, 0.4, 0.6, 0.8, 1])

    # 2nd Plot showing the actual clusters formed
    colors = cm.nipy_spectral(cluster_labels.astype(float) / n_clusters)
    ax2.scatter(x[:, 0], x[:, 1], marker='.', s=100, lw=0, alpha=0.7,
                c=colors, edgecolor='k')

    # Labeling the clusters
#    centers = clusterer.cluster_centers_
#    # Draw white circles at cluster centers
#    ax2.scatter(centers[:, 0], centers[:, 1], marker='*',
#                c="red",label='cluster 1', alpha=1, s=200,edgecolor='k')
#
#    for i, c in enumerate(centers):
#        ax2.scatter(c[0], c[1], marker='$%d$' % i, alpha=1,
#                    s=50, edgecolor='k')

    ax2.set_title("The visualization of the clustered data")
#    ax2.set_xlabel("Feature space for the 1st feature")
#    ax2.set_ylabel("Feature space for the 2nd feature")

    plt.suptitle(("Silhouette analysis for KMeans clustering on sample data "
                  "with n_clusters = %d" % n_clusters),
                 fontsize=14, fontweight='bold')

plt.show()
```

Code for persistent homology Coding Persistent Homology (Adamawa Jan1970)

1) call library for Topological Data Analysis **Jan1970**
library(TDA)

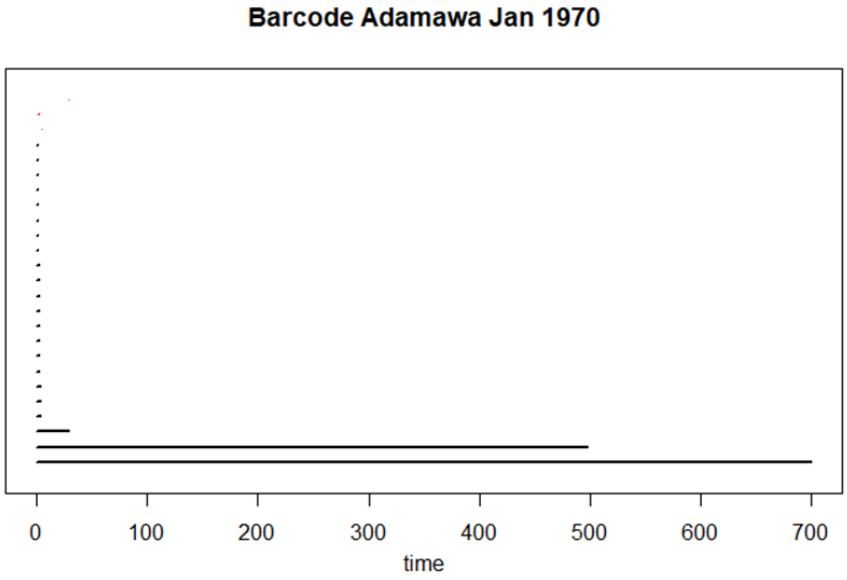
2) import Data(depend on location of the data)
ada=read.csv(file.choose())
View(ada)

3) Fix maximum filtration value (DiagLim <- 700) and maximum dimension for features (maxdimension <- 1) : means for 0-dimensional and 1-dimensional features.
DiagLim <- 700
maxdimension <-1

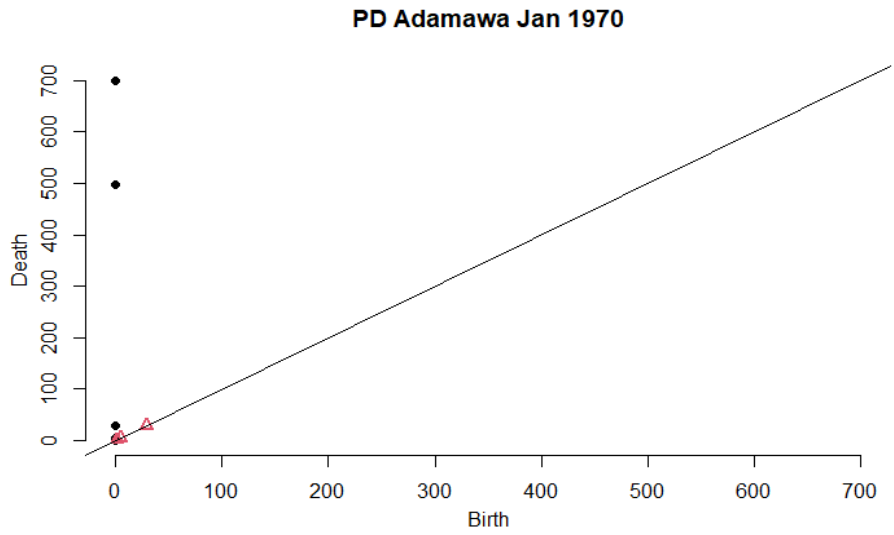
4) Apply persistent homology using Rips complex
ada1970 <- ripsDiag(ada,maxdimension,DiagLim,printProgress = TRUE)

5) Plot Barcode and Persistent Diagram (PD)

plot(ada1970[["diagram"]],barcode=TRUE,main="Barcode Adamawa Jan 1970")



plot(ada1970[["diagram"]],main="PD Adamawa Jan 1970")



6) Now calculate summary statistic of topological features sum, average, maximum lifetime and maximum death point
7) save ada1970 as matrix
matrixada1970=as.data.frame.matrix(ada1970[["diagram"]])

Example :
> matrixada1970

	dimension	Birth	Death
1	0	0.000000	700.000000
2	0	0.000000	497.116687
3	0	0.000000	28.879058
4	0	0.000000	3.464102
5	0	0.000000	3.162278
6	0	0.000000	3.162278
7	0	0.000000	2.449490
8	0	0.000000	2.449490
9	0	0.000000	2.449490
10	0	0.000000	2.236068
11	0	0.000000	2.236068
12	0	0.000000	2.236068
13	0	0.000000	2.000000
14	0	0.000000	1.732051
15	0	0.000000	1.414214
16	0	0.000000	1.414214
17	0	0.000000	1.000000
18	0	0.000000	1.000000
19	0	0.000000	1.000000
20	0	0.000000	1.000000
21	0	0.000000	1.000000
22	0	0.000000	1.000000
23	1	4.242641	4.898979
24	1	1.414214	1.732051
25	1	28.948230	29.017236

<u>calculate topological features for dimension=0 (connected component)</u>
1) First separate matrixada1970 for dimension 0 only
a=matrixada1970$dimension==0
ada1970dim0=matrixada1970[a,]

2) Second calculate lifetime of all features. Here lifetime mean the
difference between death and birth points.

ada1970dim0$difference=ada1970dim0$Death-ada1970dim0$Birth
>ada1970dim0

	dimension	Birth	Death	difference
1	0	0	700.000000	700.000000
2	0	0	497.116687	497.116687
3	0	0	28.879058	28.879058
4	0	0	3.464102	3.464102
5	0	0	3.162278	3.162278
6	0	0	3.162278	3.162278
7	0	0	2.449490	2.449490
8	0	0	2.449490	2.449490
9	0	0	2.449490	2.449490
10	0	0	2.236068	2.236068
11	0	0	2.236068	2.236068
12	0	0	2.236068	2.236068
13	0	0	2.000000	2.000000
14	0	0	1.732051	1.732051
15	0	0	1.414214	1.414214
16	0	0	1.414214	1.414214
17	0	0	1.000000	1.000000
18	0	0	1.000000	1.000000
19	0	0	1.000000	1.000000
20	0	0	1.000000	1.000000
21	0	0	1.000000	1.000000
22	0	0	1.000000	1.000000

3) Third, calculate the topological features.
4) For dimension 0, we don't consider feature with death point = 700 since it is the higher dimensional complex. All data will have one higher dimensional complex and we omit this features.
>sum(ada1970dim0$difference)-DiagLim
[1] 562.4016

> avg=(sum(ada1970dim0$difference)-DiagLim)/(nrow(ada1970dim0)-1)
>avg
[1] 26.78103

<u>calculate topological features for dimension=1 (holes)</u>
1) First separate matrixada1970 for dimension 1 only
b=matrixada1970$dimension==1

ada1970dim1=matrixada1970[b,]

2) Second calculate lifetime of all features. Here lifetime mean the
difference between death and birth points.

ada1970dim1$difference=ada1970dim1$Death-ada1970dim1$Birth
> ada1970dim1

	dimension	Birth	Death	difference
31	1	1.2893	1.52370	0.234333
32	1	21.1459	21.3577	0.211765
33	1	0.9843	1.13690	0.152519

3) Third, calculate the topological features.
>sum(ada1970dim1$difference)
[1] 1.043183
>avg=sum(ada1970dim1$difference)/nrow(ada1970dim1)
>avg

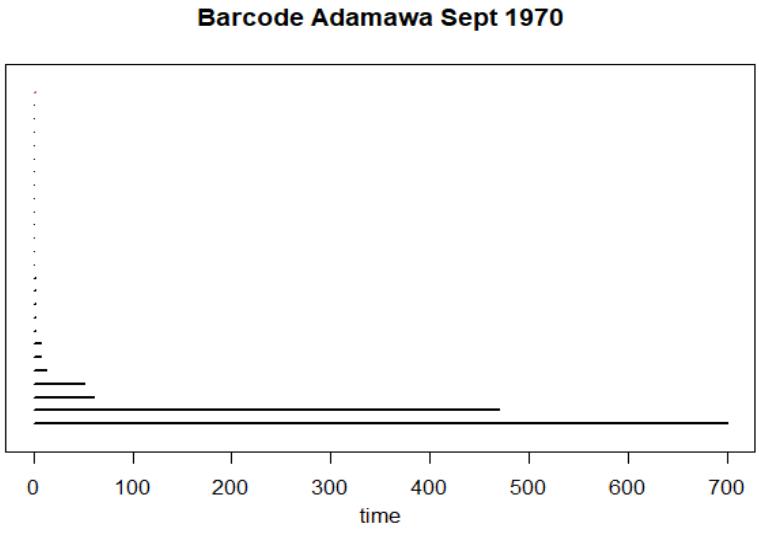
[1] 0.3477275
>max(ada1970dim1$difference)
[1] 0.6563388
>max(ada1970dim1$Death)
[1] 29.01724
Notes: Repeat all step to calculate topological features for Sept1970.
##
for Adamawa Sept1970
1) call library for Topological Data Analysis **Sept1970**
library(TDA)

2) import Data(depend on location of the data)
adam=read.csv(file.choose())
View(adam)

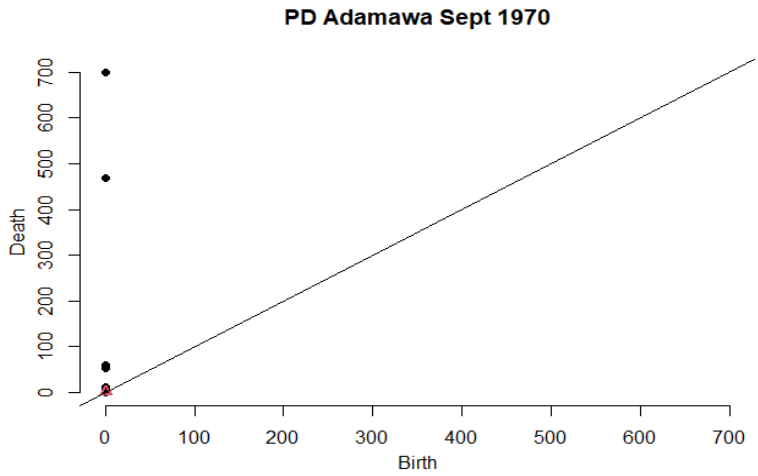
3) Fix maximum filtration value (DiagLim <- 700) and maximum dimension for features (maxdimension <- 1) : means for 0-dimensional and 1-dimensional features.
DiagLim <- 700
maxdimension <-1

4) Apply persistent homology using Rips complex
adam1970 <- ripsDiag(adam,maxdimension,DiagLim,printProgress = TRUE)

5) Plot Barcode and Persistent Diagram (PD)
plot(adam1970[["diagram"]],barcode=TRUE,main="Barcode Adamawa Sept 1970")



plot(adam1970[["diagram"]],main="PD Adamawa Sept 1970")



6) Now calculate summary statistic of topological features sum, average, maximum lifetime and maximum death point
7) save adam1970 as matrix
matrixadam1970=as.data.frame.matrix(adam1970[["diagram"]])

Example :
> matrixadam1970

dimension	Birth	Death
1	0	0 700.000000
2	0	0 469.171610
3	0	0 61.040970
4	0	0 51.874849
5	0	0 12.727922
6	0	0 7.141428
7	0	0 7.071068
8	0	0 2.236068
9	0	0 1.414214
10	0	0 1.414214
11	0	0 1.414214
12	0	0 1.414214
13	0	0 1.000000
14	0	0 1.000000
15	0	0 1.000000
16	0	0 1.000000
17	0	0 1.000000
18	0	0 1.000000
19	0	0 1.000000
20	0	0 1.000000
21	0	0 1.000000
22	0	0 1.000000
23	0	0 1.000000
24	0	0 1.000000
25	0	0 1.000000
26	1	1 1.414214

calculate topological features for dimension=0 (connected component)
1) First separate matrixadam1970 for dimension 0 only
a=matrixadam1970$dimension==0
adam1970dim0=matrixadam1970[a,]

2) Second calculate lifetime of all features. Here lifetime mean the difference between death and birth points.

adam1970dim0$difference=adam1970dim0$Death-adam1970dim0$Birth
>adam1970dim0

	dimension	Birth	Death	difference
1	0	0	700.000000	700.000000
2	0	0	469.171610	469.171610
3	0	0	61.040970	61.040970
4	0	0	51.874849	51.874849
5	0	0	12.727922	12.727922
6	0	0	7.141428	7.141428
7	0	0	7.071068	7.071068
8	0	0	2.236068	2.236068
9	0	0	1.414214	1.414214
10	0	0	1.414214	1.414214
11	0	0	1.414214	1.414214
12	0	0	1.414214	1.414214
13	0	0	1.000000	1.000000
14	0	0	1.000000	1.000000
15	0	0	1.000000	1.000000
16	0	0	1.000000	1.000000
17	0	0	1.000000	1.000000
18	0	0	1.000000	1.000000
19	0	0	1.000000	1.000000
20	0	0	1.000000	1.000000
21	0	0	1.000000	1.000000
22	0	0	1.000000	1.000000
23	0	0	1.000000	1.000000
24	0	0	1.000000	1.000000
25	0	0	1.000000	1.000000

3) Third, calculate the topological features.
4) For dimension 0, we don't consider feature with death point = 700 since it is the higher dimensional complex. All data will have one higher dimensional complex and we omit this features.
>sum(adam1970dim0$difference)-DiagLim
[1] 629.9208

> avg=(sum(adam1970dim0$difference)-DiagLim)/(nrow(adam1970dim0)-1)
>avg
[1] 26.2467

calculate topological features for dimension=1 (holes)

1) First separate matrixadam1970 for dimension 1 only
b=matrixadam1970$dimension==1

adam1970dim1=matrixadam1970[b,]

2) Second calculate lifetime of all features. Here lifetime mean the difference between death and birth points.

adam1970dim1$difference=adam1970dim1$Death-adam1970dim1$Birth
> adam1970dim1

	dimension	Birth	Death	difference
26	1	1	1.414214	0.4142136

3) Third, calculate the topological features.
>sum(adam1970dim1$difference)
[1] 0.4142136
>avg=sum(adam1970dim1$difference)/nrow(adam1970dim1)
>avg
[1] 0.4142136
>max(adam1970dim1$difference)
[1] 0.4142136

```
>max(adam1970dim1$Death)
[1] 29.01724
```