Noname manuscript No.
(will be inserted by the editor)

Metric space change point detection

David Letscher · Darrin Speegle

Received: date / Accepted: date

Abstract Let $(x_t)_{t=1}^N$ be a time series with values in a metric space X, which is locally isometric to Euclidean space. A transformation of the data is proposed which produces a multi-dimensional time series of real numbers. If the original sequence of data points has a single change point in mean at time t_0, then with high probability the transformed data will also have a single change point in mean at time t_0. Applications to time series of persistence diagrams are considered.

Keywords change point detection · metric space · persistence · tda

Mathematics Subject Classification (2000) MSC code 1 · MSC code 2

1 Introduction

A simple change point detection problem is the following: suppose that $(x_t)_{t=1}^{t_0}$ are iid Gaussian random variables with mean μ_0 and variance σ, and $(x_t)_{t=t_0+1}^N$ are iid Gaussian with mean μ_1 and variance σ. The goal is to determine whether $\mu_0 = \mu_1$, and if not, then what the value of t_0 is.

More general problems include having multiple change points, noise that is not Gaussian, underlying signals that are not constant, and multi-dimensional signals.

Grants or other notes about the article that should go on the front page should be placed here. General acknowledgments should be placed at the end of the article.

David Letscher
Department of Computer Science, Saint Louis University
E-mail: david.letscher@slu.edu

Darrin Speegle
Department of Mathematics and Statistics, Saint Louis University
E-mail: darrin.speegle@slu.edu

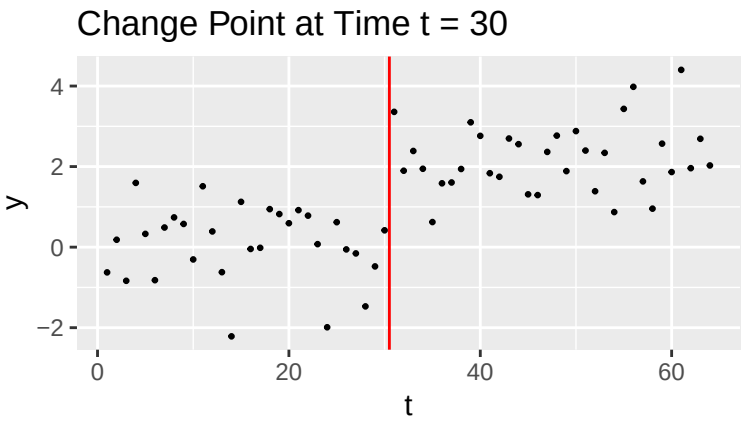

Fig. 1 Simple change point

In this paper, we consider the following set-up. Let X be a metric space. Supose that $f : [0,1] \to X$ is a piecewise continuous function with at most one discontinuity. Let $(a_t)_{t=1}^N$ be an increasing sequence of numbers in $(0,1)$ and $\delta > 0$ such that for each $y \in [0,1]$, $B_\delta(f(y))$ is isometric to Euclidean space. Let $x_t = f(a_t) + \epsilon_t$, where ϵ_t is a uniformly distributed random variable on $B_\delta(f(a_t))$. Our problem is to determine whether there exists a point of discontinuity y_0 of f and a $k < N$ such that the discontinuity is contained between a_k and a_{k+1}. If there does exist such a k, then we should also estimate it.

Our motivation is that we wish to find change points in time series of **persistence diagrams**. Persistence diagrams are a way of measuring the topological structure of a point cloud in Euclidean space. Given a collection of points $\{x_1, \ldots, x_M\} \subset \mathbb{R}^N$ and an $\alpha > 0$, we create a Rips complex that consists of all subsets of $A \subset \{x_1, \ldots, x_M\}$ that have the property that $\max\{d(x,y) : x, y \in A\} < \alpha$. For each $\alpha > 0$, we then compute the simplicial homology of the Rips complex. Features in the simplicial homology can be associated with a birth time (the smallest α for which the feature appears) and a death time (the largest *alpha* for which the feature appears). Inituitively, features that have a large difference between birth and death time are more likely to be intrinsic to the point cloud than features that only persist over a small interval.

For example, consider the collection of points in $\mathbb{R}^2$ shown below. A .gif showing the Rips filtration of the point clouds can be found here. There are 18 connected components when α is very small, and as α increases, there are fewer and fewer until finally at $\alpha = 0.8$, there is only one. A loop forms when $\alpha = 1$ and persists until $\alpha = 1.4$. These facts are demonstrated in Figure 2.

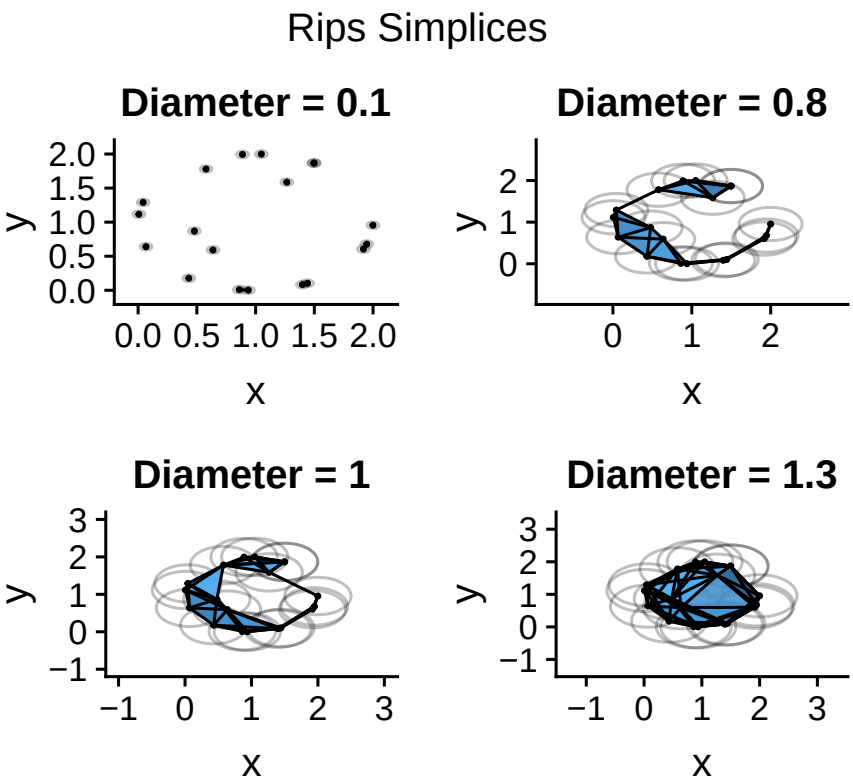

Fig. 2 Important diameters of the Rips filtration of a point cloud sampled from a circle

Based on Figure 2, we would expect that three components would have death time less than 0.1, that all components except one wil have died by time 0.8, that a loop will appear before time 1, and that the loop will die before time 1.3.

Using the `ripsDiag` function of the `TDA` R package, we can see that, indeed, the first three death times are 005, 0.08 and 0.08, the last death time of a connected component is 0.73, and the birth and death times of the loop are 0.97 and 1.28, respectively. The full persistence diagram is given in Table 1.

Table 1: Birth and death times of zero dimensional and one dimensional features.

dimension	Birth	Death
0	0.00	5.00
0	0.00	0.73
0	0.00	0.69

0	0.00	0.48
0	0.00	0.47
0	0.00	0.47
0	0.00	0.47
0	0.00	0.46
0	0.00	0.46
0	0.00	0.38
0	0.00	0.36
0	0.00	0.32
0	0.00	0.28
0	0.00	0.18
0	0.00	0.16
0	0.00	0.08
0	0.00	0.08
0	0.00	0.05
0	0.00	0.00
1	0.97	1.28

We will use the bottleneck or Wasserstein distances on persistence diagrams. We refer to [**?**] for details.

2 Metric space change point detection algorithm

With the notation as set up in Section 1, we proceed to describe the algorithm for change point detection for time series data in metric spaces. The main step is transforming the data into a multi-dimensional time series of real numbers in such a way as to preserve the change point, see Algorithm 1.

Algorithm 1 Transform to Real

```
1: procedure MyProcedure
2:     time_length ← length of time_series
3:     i ← 1
4:     while i < time_length do
5:         A, B ← sample(N, 2, replace = FALSE)
6:         for j ← 1 : time_length do
7:             dists[j,i] ← (d(A, x_j)^2 − d(B, x_j)^2)/d(A, B)
8:         i++
```

Once the metric space valued time series is transformed into a multi-dimensional real valued time series, standard techniques can be used to determine if and where a change point occurs. In this paper, we use the `cpbaywave` package, as it finds change points in high dimensional, smooth data with a single discontinuity. We illustrate the usage with some examples.

Example 1 Suppose that the time series lives in $\mathbb{R}^M$ for some M. For example, we could have a time series of length 128 in 100 dimensional data, with a change point at time $t = 80$. We consider two examples in this case; in both examples, the data has mean zero in all dimensions until time 80. In the first case, it has mean 0.1 in all dimensions from dimension 81 through 128 and in the second case it has mean 1 and all dimensions from dimension 81 through 128. In all cases, the standard deviation is 1.

As shown in Figure 3, in the case that the mean increases by 0.1, the algorithm is not able to locate the change point, and incorrectly determines that there is no change. We note that `cpbaywave` was also unable to detect the change point when using the untransformed data.

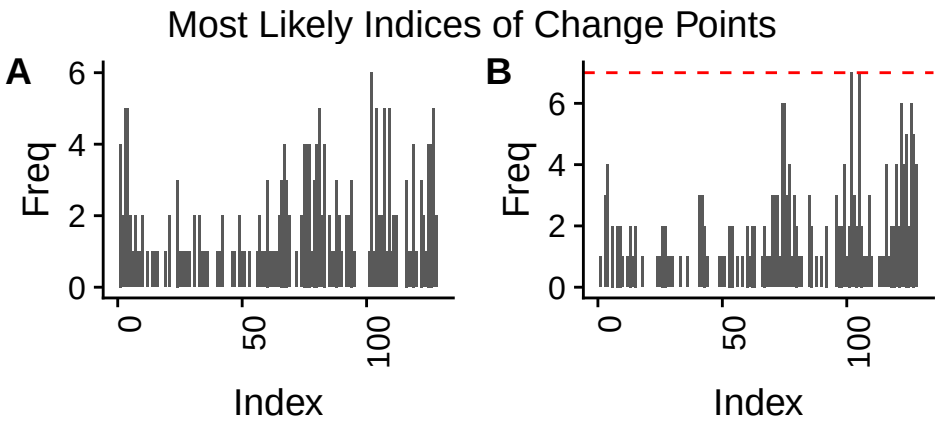

Fig. 3 Plot A is from raw data, and plot B is from transformed data. In neither case is the change point detected

However, as shown in Figure 4 and Table 2, the algorithm is able to correctly identify the change point of $t = 80$ when the mean increases by 1. We note that `cpbaywave` was also able to detect the change point using the untransformed data.

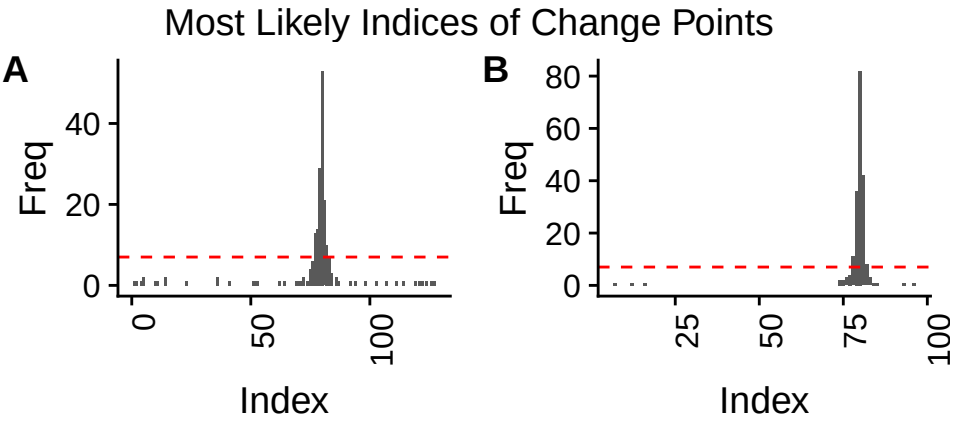

Fig. 4 Plot A is from raw data, and plot B is from transformed data. In both cases, the change point is detected.

Table 2: Indices which are significant, together with their frequency. True change point at t = 80.

Raw Data		Transformed Data	
Index	Freq	Index	Freq
77	13	78	11
78	14	79	36
79	29	80	82
80	53	81	42
81	21	82	8
82	10		

This picture gives a typical strong indication that there is a change point at or near time $t = 80$, which we know is the correct value.

Example 2 Next, we consider a change in mean from $\mu = 0$ to $\mu = 1$ at time $t = 80$, as above, but we imagine the time series living in ℓ_4^{100} rather than in ℓ_2^{100}. Since ℓ_4^{100} is not locally Euclidean, this example is testing the robustness of our algorithm to our assumptions. As can be seen in Figure 5 and Table 3, the change point is detected in the raw data (which we had already seen in Example 1) as well as the transormed data. Thus, our algorithm appears to have some robustness to non-Euclidean metric spaces.

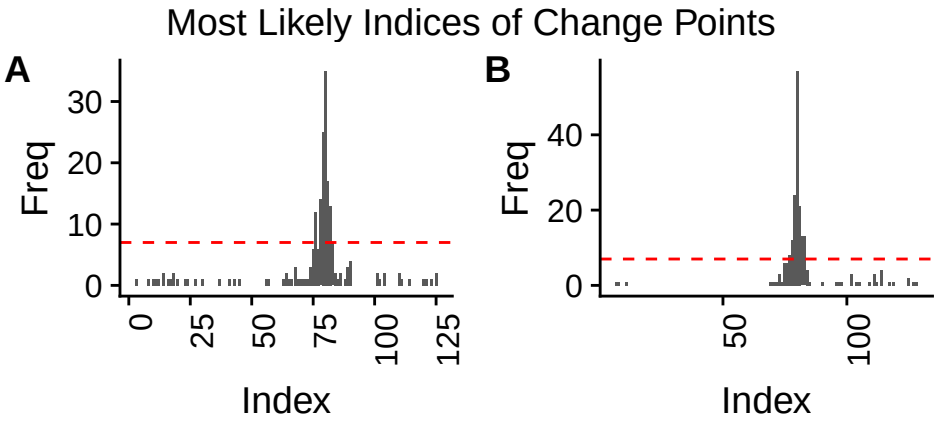

Fig. 5 Plot A is from raw data, and plot B is from transformed data, assuming that the data lives in ℓ^4. In both cases, the change point is detected.

Table 3: Indices which are significant, together with their frequency.

Raw Data		Transformed Data	
Index	Freq	Index	Freq
76	12	77	8
78	14	78	12
79	25	79	24
80	35	80	57
81	17	81	21
82	13	82	13
		83	13

3 Change points in persistence diagrams

The space of persistence diagrams is not locally isometric to Euclidean space. However, it does seem to be closer to being isometric to Euclidean space than ℓ^4 is. We apply our algorithm now for time series of persistence diagrams. First, we apply it to simulated data, then to data in the wild of various types.

Example 3 Our first example is of two time series of simulated point clouds with strong topological properties. For the first time series, we start with a single circle, sampled randomly with normal jitter, and then at time $t = 80$, we add a second point cloud sampled from a disjoint circle with normal jitter added. Any good change point detection algorithm based on persistence diagrams should be able to find such a change. Figure 6 shows time samples before and after the change point.

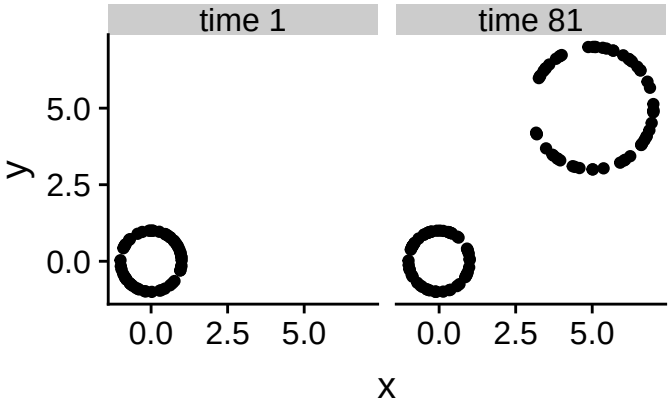

Fig. 6 Point clouds before and after the change point, where a second circle is added.

The change point detection algorithm easily detects the change, as seen in Figure 7.

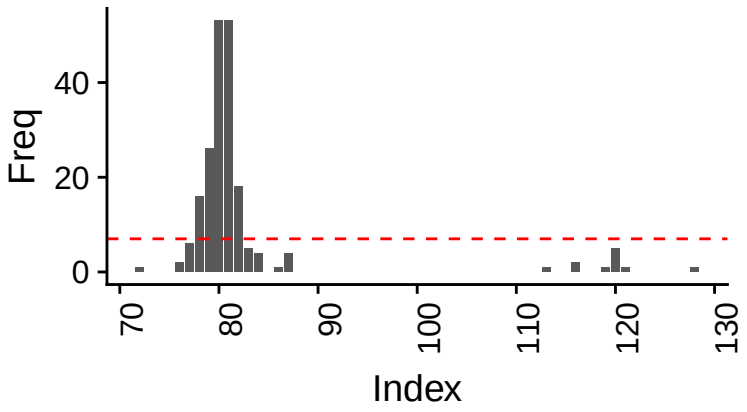

Fig. 7 Change point detected when second circle is added at time $t = 80$.

As seen in Table 4, times 80 and 81 were the most commonly chosen times for the change point in the bootstrap algorithm, with 78, 79 and 82 also making the cutoff to be significant.

Table 4: Indices which are significant, together with their frequency. True change point at t = 80.

Index	Freq
78	16
79	26
80	53

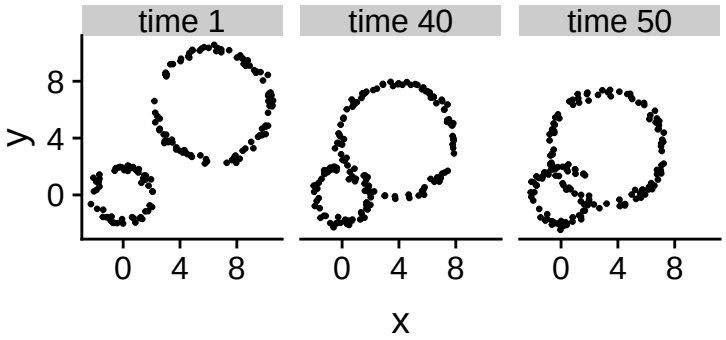

Fig. 8 Circles move through each other. Topological change point not clearly defined.

81	53
82	18

Our second example of simulated data is more challenging. We sample from two circles (again with normal jitter) which start out being disjoint, but over time one of the two circles moves so that it intersects the other circle. From a topological point of view, there are two potential change points. The first change point is when there are no longer two connected components, but rather one. The second change is when there are three loops rather than two. The algorithm currently under discussion only finds a single change point; how many and which dimensions are included in the persistence diagram and distance calculation will determine which change is detected.

Figure 8 shows the point clouds at three time points, which illustrate the three different (topological) states that the point couds may have. See also here for a .gif showing the entire time series.

In Figure 9, we see that the change point detection algorithm detects approximately when three distinct loops become show in the time series. Note that in this case, we only used 1-dimensional homology in our distance calculation, so our algorithm would not be able to detect changes in clusters.

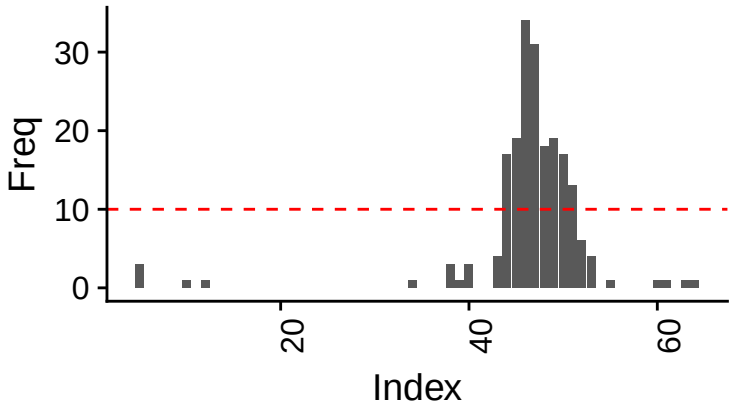

Fig. 9 When two circles move through each other, a topological change is detected when the third loop becomes prominent, at or about time $t = 46$.

Finally, we see in Table 5 that 8 different indices are significant via the bootstrap, with the most likely candidate being time $t = 46$.

Table 5: Indices which are significant, together with their frequency. True change point ambiguous.

Index	Freq
44	17
45	19
46	34
47	31
48	18
49	19
50	17
51	13

Now, we turn to image data. Our basic algorithm is to take a time series of images, find the edges at each time, sample from the edges at each time to form a time series of point clouds, then follow Algorithm 2. We illustrate this with two examples. In one example, we look at pictures of Table Mountain over time, and it is not completely clear whether or when there is a change point in the time series. In the other example, we examine pictures of a cell dividing, in which case there is a clear change in (topological) characteristics of the images at the time of cell division.

Example 4 In our first example, we use data from the Archive of Many Outdoor Scenes. We picked a camera that was taking pictures of Table Mountain in South Africa. For each day, we averaged all of the pictures that were taken on that day. We then used edge detection to extract edges from the pictures,

and we removed the time stamp. Finally, we randomly sampled points from the detected edges of the pictures to use as our point cloud. So, the time series consists of randomly sampled points from edges of the average of all pictures on 64 consecutive days. The original averaged images are here and the sampled edges are here. Note that these time series are longer than length 64; we only used the first 64 images in our algorithm to avoid complications of padding.

In Figure 10, we see that no change point is detected in the time series, though one of the index 63 did beat the nominal bootstrap line.

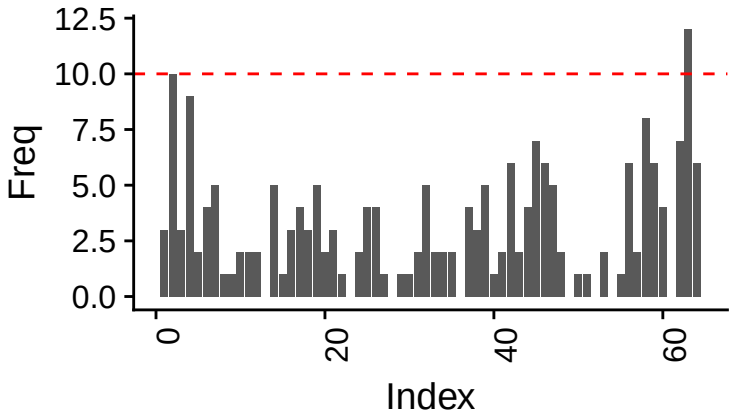

Fig. 10 No significant change point found in the Table Mountain images.

Example 5 In our last example, we have a series of images of a cell, which divides at about time $t = 23$. See here here for the raw data, We again used the `cannyEdges` function in the `imager` R package to detect the edges of the cells. We then randomly sampled 200 points from the detected edges at each time stamp to form our time series of point clouds, see here for the results. See Figure 11 for typical pictures before and after the change point. We then applied Algorithm 2 to detect change points.

Fig. 11 Raw image of cells before and after dividing, together with edge detection.

In Figure 12, we see good evidence of a change near the time $t = 23$, which was estimated to be the correct time by the authors of this paper.

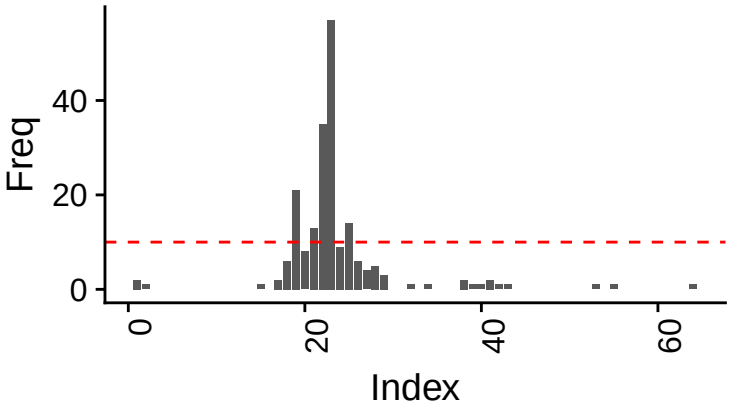

Fig. 12 Detects change at or around time $t = 23$.

Table 6 shows the significant indices from the bootstrap. There are more than would be ideal, but the change points do at least tend to cluster around the correct value, indicating that there is something of interest at or near those points.

Table 6: Indices which are significant, together with their frequency.
True change point at time t = 23

Index	Freq
19	21

21	13
22	35
23	57
25	14

4 References