Introduction to Change Point Analysis

Ryan Porter[1]

[1] *Department of Mathematics, Boise State University, Boise, Idaho 83725*

May 7, 2019

Abstract

One of the biggest challenges in time series and sequence data is the ability to detect multiple points of changes within that data set. Change point analysis is the process of detecting multiple points of changes throughout time series or sequence data. There are many different change point search methods but for this paper the focus will be on PELT and Binary Segmentation methods. The different search methods and demonstrations of their application with simulated and real life data sets will be explored throughout the paper. A change point package has been developed by Rebecca Killick [8] that incorporates these methods in R which will be used for our research.

KEY WORDS: Change Point; AMOC; PELT; Binary Segmentation.

1 Introduction

Suppose that a scientist is curious about the changes in annual water flow volume of the Nile River in a given range of about 100 years. The scientist is curious as to which year had the biggest changes in water flow compared to the rest of the data. He finds one point in the data set and determines that this must be the most important year since it had the biggest change in mean. However, what if there was another year that had a change in mean with a very similar magnitude as the one that he found? What if there are multiple years that all have about the same magnitude change in mean as the one found? This is where change point analysis comes in it is the idea of finding multiple points of change in time series or sequence data.

There is an ever growing need to be able to identify multiple change points within a time series or sequence data. Getting the location of each of the change points is imperative in order to get the most accurate analysis of the data possible. However, with data sets increasing in size rapidly in the last few years so does the number of potential points of change.

The equation (1) shows a commonly used approach to identify multiple change points, which is to minimize

$$\sum_{i=1}^{m+1}[C(y_{(\tau_{i-1}+1)})] + \beta f(m) \tag{1}$$

In the equation, C represents the cost function for the segment and $\beta f(m)$ is the penalty to stop if from over-fitting. m is the number of change points and τ is the change point position. However, the penalty for change point analysis will not be discussed in this paper, so default penalty will be used. Even though penalty the different penalty methods will not be used, some common methods are Akaike Information Criterion (AIC), the Schwarz Information Criterion (SIC), and Modified Bayes Information Criterion (MBIC) [5].PELT and Binary Segmentation methods try to achieve the smallest cost function in order to compute the change points as efficiently as possible [4]. The next section will give more background into the two different change point methods that were used in this research.

2 Background

2.1 PELT

PELT, Pruned Exact Linear Time, is the method of identifying change points in a time series data set using an exact search. Normal exact search methods have a high computational cost but PELT is able to do exact search with lower computation. PELT is achieves exact search faster than other methods of exact search, especially for large number of data points. However, it has been observed that PELT is slower than other methods of relative search like Binary Segmentation.

The PELT equation is similar to equation (1) but with pruning of the data. The Pelt method combines optimal partitioning and pruning which results in exact and efficient search method. Thus we get the following equation:

$$F(n) = \overset{min}{\tau_m} [F(\tau_m) + C(y_{(\tau_{i-1}+1),...,y_n})] \tag{2}$$

First, start by calculating $F(1)$ and then recursively calculate $F(2), ..., F(N)$. At every step we store the optimal segmentation up to τ_{m+1}. When the equation reaches $F(n)$, the optimal segmentation for the entire set will be identified and the number with the location of each change point has been recorded. The efficiency of PELT is achieved by removing candidate values of τ_m from the minimisation at each step [5].

A downfall of using an exact search method is the amount of time it takes for the computation to take place. This might not be an important problem when working with smaller data sets but what about the data sets that have large amount of change points? Research done by Killick and Fearnhead showed that in the worst case scenario of complexity of the PELT algorithm is where no pruning occurs, which causes the computational cost to be $\mathcal{O}(n^2)$. Segment Neighborhood [6], which is another exact search method, has the computational cost of $\mathcal{O}(n^3)$ which is significantly larger than PELT. If the number of change points were linearly increasing then PELT would have an even smaller computational cost than worst case scenario, $\mathcal{O}(n^2)$.

2.2 Binary Segmentation

Binary Segmentation is the most established search method that is used in change point analysis. This method essentially takes a single change point method and repeats that method over and over on different subsets in a sequence. To build off that it initially applies the single change point method to the entire set of data, tests if τ exists that satisfies

$$C(y_{1:\tau}) + C(y_{(\tau+1):n}) + \beta < C(y_{1:n}) \tag{3}$$

If equation (3) is false then there is no change point and the method stops running. If the equation return a true value, then it splits the data set at the detected change point, and the method is now ran on the two subsets of data. This continues on until it returns back a false value for the equation (3).

The computational cost of Binary Segmentation is different than PELT since Binary Segmentation is not an exact search method [6]. Binary Segmentation has approximate computational cost to $\mathcal{O}(n \log n)$, where n is the number of data points. However, to achieve such efficiency there is no guarantee that it will find the global minimum of the data set. Since adds one change point at a time, they are added in the position that leads to the largest reduction in cost given the location of the previous change points [6]. Thus, the use of this approximation of the solution leads to it being more efficient but less accurate than PELT.

2.3 R Package

There are R packages that have already been developed for change point analysis which was useful for this experiment. The package developed by Rebecca Killick "changepoint" [7] was the one used for the testing. In this package new classes were introduced but the primary one being *cpt.mean* function:

```
cpt.mean(data, penalty = "Manual", pen.value = 0, method = "AMOC", Q = 1, test.stat
= "Normal", class = TRUE, param.estimates = TRUE)
```

- data - A vector containing the data within which to find a change in mean
- penalty - Choice of "None", "SIC", "BIC", "AIC", "Hannan-Quinn", "Asymptotic", and "Manual" penalties.
- pen.value - When using a "Manual" penalty it is a numeric value.
- method - Single or multiple change point method, either "AMOC", "PELT", "SegNeigh" or "BinSeg".

- Q - Maxium number of change point or segments, only applies to ”BinSeg” and ”SegNeigh”.

  test.stat - The test statistic, either ”Normal” or ”CUMSUM”

- class - Logical, if true then the class cpt is returned

- param.estimates - Logical, if true and class true then the parameter estimates are returned.

The package includes two different functions *cpt.mean* and *cpt.var* but since I only tested changes in mean in the data, *cpt.mean* will be the only needed function.

3 Data

There are three different data sets that are used for examples within this paper. The first data set is simulated in order to focus more on the ideas surrounding the process rather than the data. The simulated data set is composed of 400 points that span the values from −2.5 to about 3.1. The purpose of the data is to have a lot of small changes to test if the methods can find changes in mean and how accurate the different methods are. The following code shows the creation of what will be referred to as the sim.data

```
> set.seed(10)
> sim.data <- c(rnorm(100, 0, 1), rnorm(100, 1, 1),
    rnorm(100, 0, 1), rnorm(100, 0.2, 1))
```

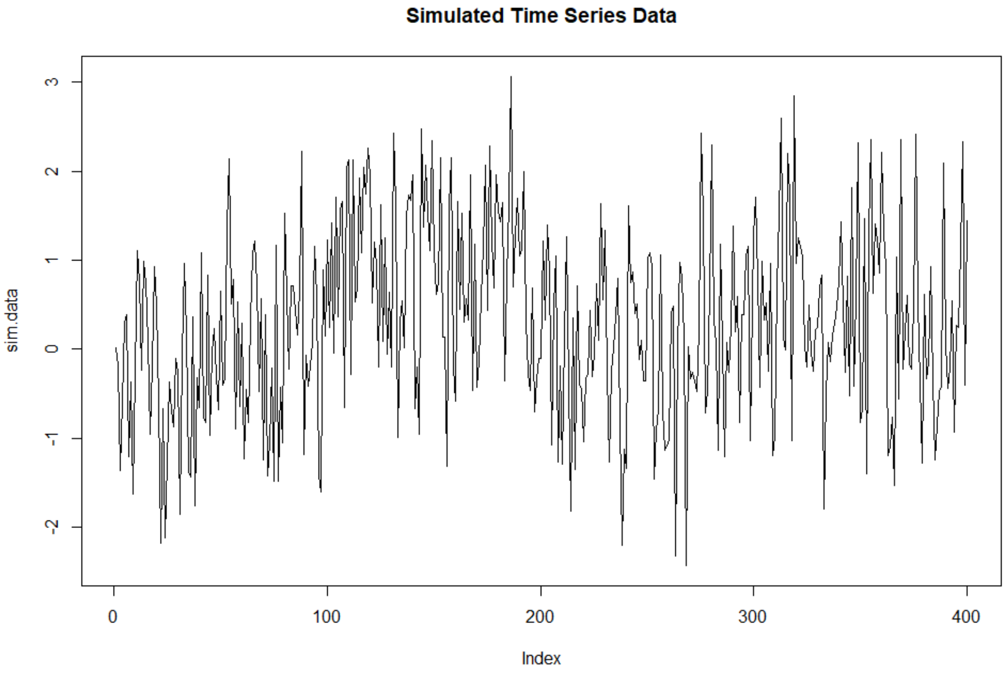


Figure 1: Simulated Data

The second data set is the annual flow volume of water for the Nile River at Aswan from 1871 to 1970. This is the data set that was talked about in the hypothetical situation in the introduction. The data set is actually part of R so you don't need to download any additional items. The reason I chose this data set was I wanted to explore if the change points correlated to important dates in the history of the Nile River. To load in the data about the Nile River follow the code below:

```
> data(Nile)
> dev.new()
> ts.plot(Nile)
```

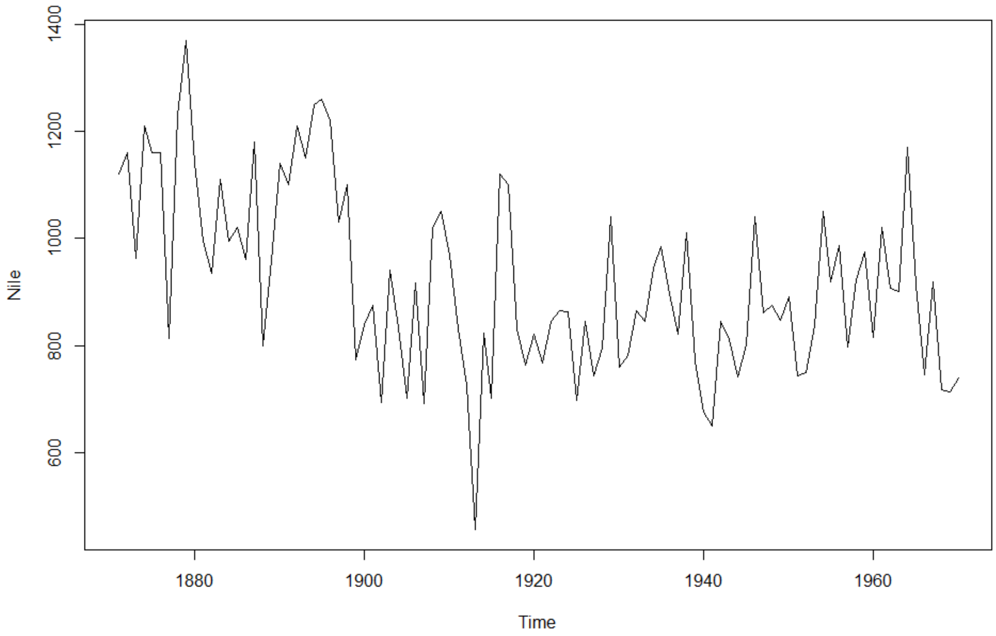


Figure 2: Nile River Data

The final data set is a time series for the heart rate of someone who is initially resting but then starts working out. This data set was collected from GitHub with the purpose of using it for analysis with 1,160 data points. The data is a bit different in the sense that there is an initial big change in the series but then transitions into more small changes. Change point analysis will be put to the real test here as there is an initial big change point which could overshadow the other smaller changes. The heart rate data is not part of a library so you'll have to manually load in the data from your system, use the following code

```
> load("D:/School/College - Senior/Thesis/HeartRate.RData")
> dev.new()
> ts.plot(HeartRate, main = "Heartbeat Data")
```

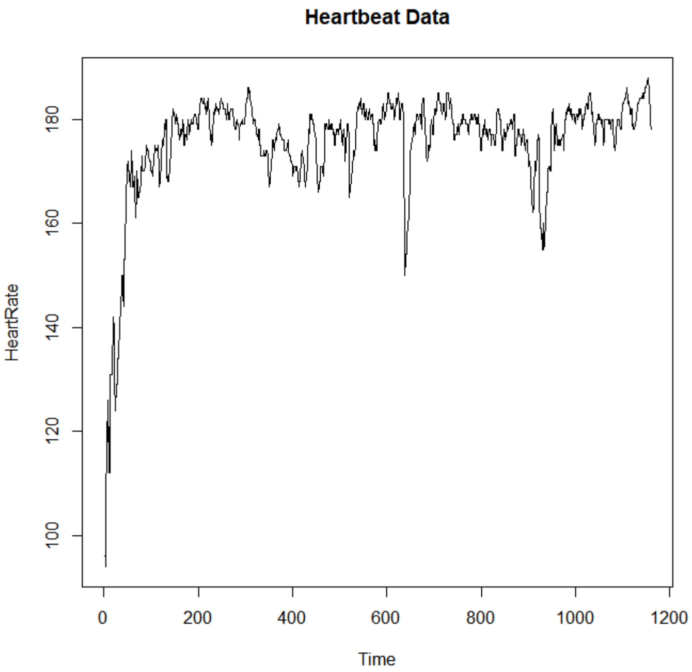


Figure 3: Heart Beat Data

4 Simulations

4.1 Single Change Point

A big challenge in analyzing data is finding the point or points in which the data has the greatest change. Before tackling the idea of having two or more change points in a single time series or sequence, lets first start by finding one change. Luckily, the changepoint package has a method called AMOC. This method stands for ”At most one change” so it only looks for one change in the time series data. There could be many changes in the data but the AMOC method will return one value. Recall the simulated data that was created earlier in data section with 400 different points in the list. That means when using the method AMOC it will only choose one point out of the 400 that has the greatest change in mean. Figure 3.1 is a plot of the simulated data which shows the variation of the points.

From just looking at Figure (1) it is pretty difficult to determine where the biggest change in the mean is for the data. There looks to be a change point around the area from 100 to 130, but no way to tell just based off the graph. Thus, the method AMOC is used in the following code to determine where the change point is.

```
> sim.amoc <- cpt.mean(sim.data, method = "AMOC")
> dev.new()
> plot(sim.amoc, type = "l", xlab = "Index", cpt.width = 4)
> cpts(sim.amoc)
```

```
[1] 79
```

It gives the result of 79 which was close to the educated guess based on the graph. The problem that analysts are now faced with, is there still seems to be a lot of variation in the latter part of the graph. What if there is another point in the graph that has almost the same amount of change in mean but is slightly smaller than the chosen point? This is where finding multiple change points would come in order to determine if there are more than just one significant point of change in mean.

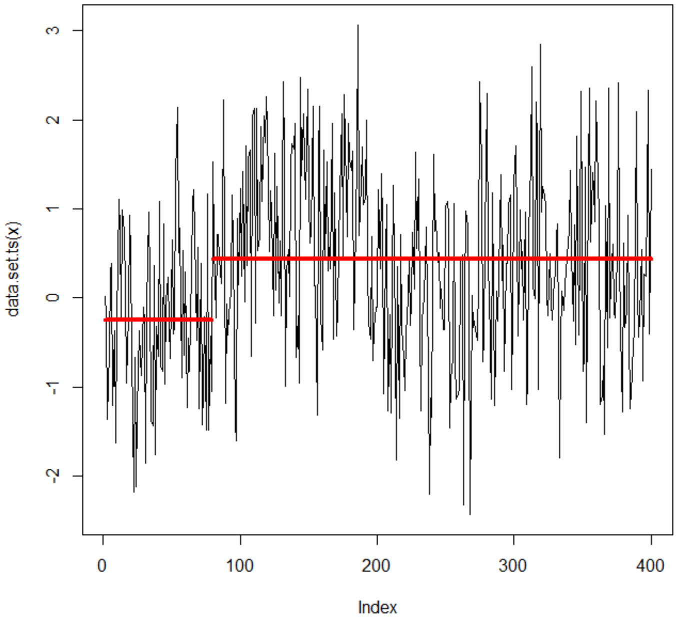


Figure 4: Simulated Data AMOC

The heart beat data that was introduced has an obvious change in mean at the beginning of the time series data. The same test that was used for the simulation data is repeated for the heart rate data using the AMOC method. In figure (3), shows the heart rate data as a time series plot and figure (5) shows the data using the AMOC method. A result of $t_0 = 44$ is produced which can be visually seen in Figure (5) at the break point between the red line. It is no surprise that the AMOC method choose the change point at the start of the time series since it was such a drastic change in mean.

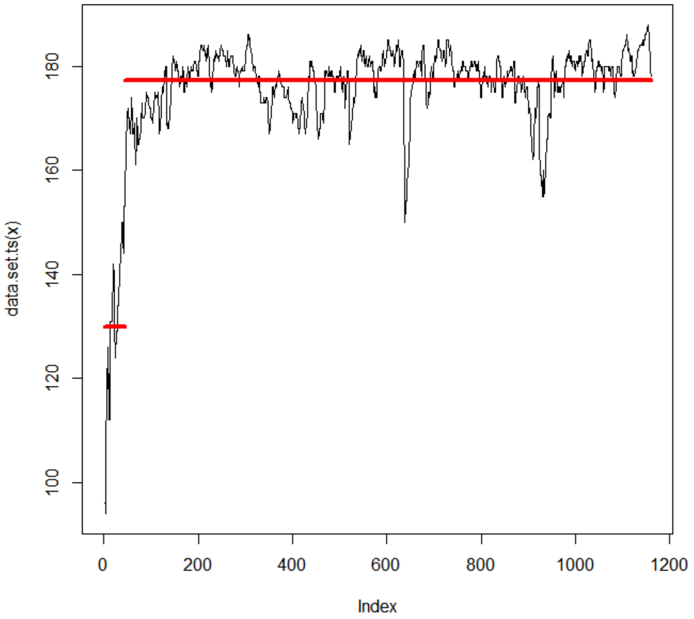


Figure 5: Heart Rate AMOC

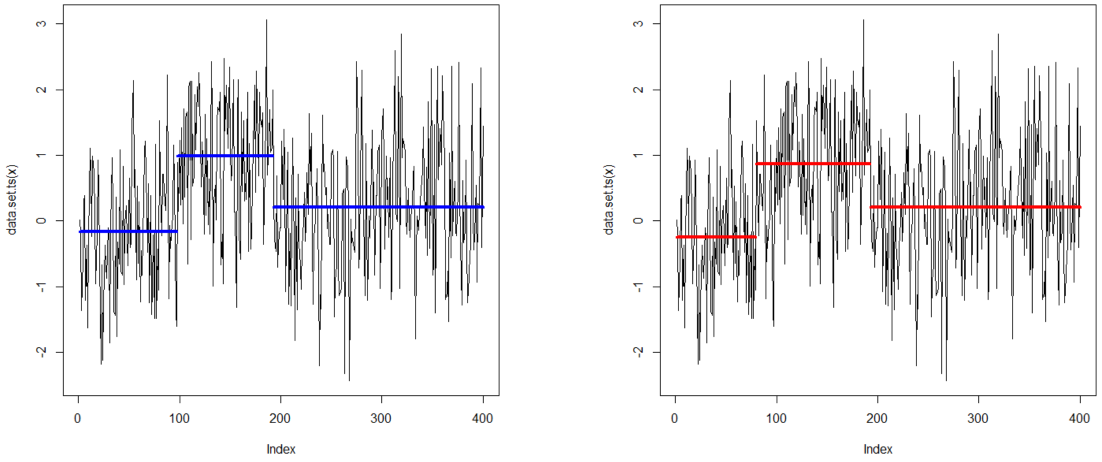
4.2 Multiple Change Points

Detecting multiple points of change in a time series graph could help analysts come to a more precise conclusion of their data. In the previous section a single change point was found using the AMOC method, but what if there are multiple points of interest in the data? Lets look again at the simulated data which seemed to have many points of change but with a method that looks for multiple change points. The code is very similar to the code of a single change point but the method has now changed

```
> sim.pelt <- cpt.mean(sim.data, method = "PELT")
> dev.new()
> plot(sim.pelt, type = "l", cpt.col = "blue", xlab = "Index", cpt.width = 4)
> cpts(sim.pelt)
[1]  97 192
```

From the results it is apparent that it gave us an additional point of change but kept the same original one as AMOC. The results are assuming default penalty (17.97439), if the method produced too many points a penalty is used to combine segments. Lets now take a look at how the PELT method compared to the Binary Segmentation method which is not an exact search method.

```
> cpts(sim.binseg)
[1]  79 192
```



(a) PELT

(b) Binary Segmentation

Figure 6

It is interesting that both PELT and Binary Segmentation choose the same second change point but not the first. Remember that PELT is an exact search method so the change points that it finds will be more accurate than Binary Segmentation. Also Binary Segmentation uses the same method just recursive for determining a change point that AMOC uses so it is no surprise that they found the same initial change point. However, it is apparent that there is a change later in the simulated data that was not captured by the AMOC method. By only using the AMOC method that could mean that the analyst is not able to full grasp the story of the data as time goes on. By using a multiple change point method it allows the data to a better story to the analyst to make more educated decisions on the data. PELT and Binary Segmentation both gave us two points but one is an exact search method and the other isn't, which efficiency might come into question. Running an exact search method usually takes more time which is why Binary Segmentation is more used in application. In R it can show the compute time it took to run both of those methods.

```
> system.time(cpt.mean(sim.data, method = "PELT"))
   user  system elapsed
      0       0       0
> system.time(cpt.mean(sim.data, method = "BinSeg"))
   user  system elapsed
   0.11    0.01    0.12
```

PELT method was so quick at computing that it is less than 0.009 seconds which then appears to be 0 in the output. On the other hand we can see that Binary Segmentation took about 0.12 seconds to compute. These are two different types of search methods which needs to be remembered when comparing the compute time between the two. I repeated this whole process for all three different methods but changing the seed number each time. The seed number was changed to $50, 100, 1000, 10000$ which the results are shown in Figure (7).

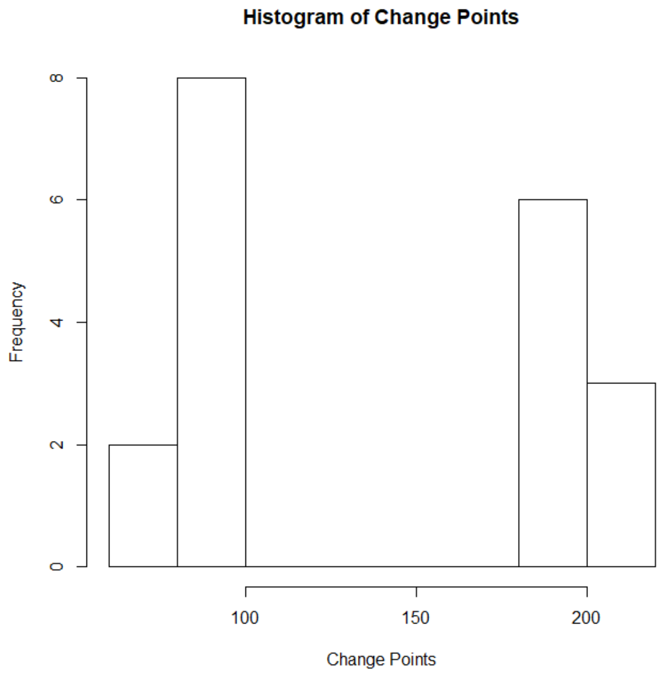
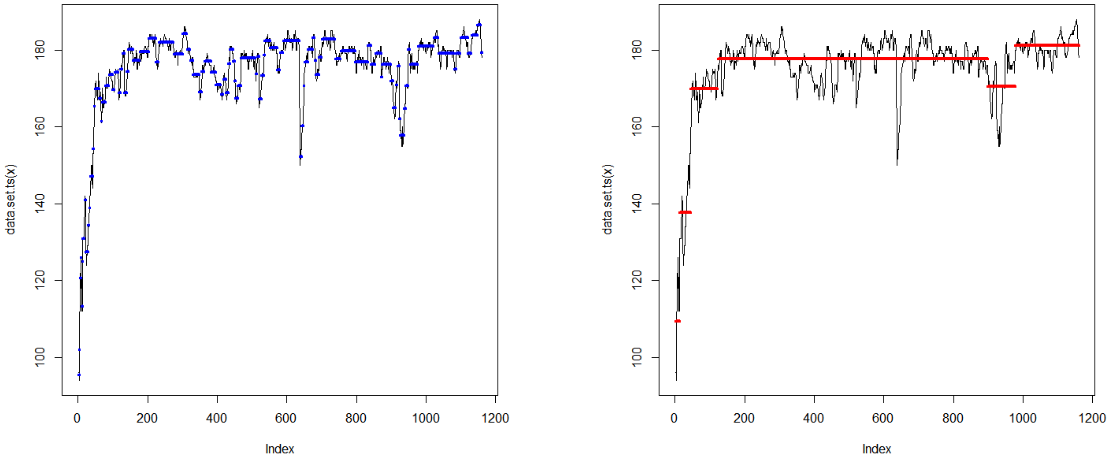


Figure 7: Histogram of Change Points

5 Applications

5.1 Heart Rate

The simulated data has shown how both methods can produce multiple points of change in a data set. However, that was on a constructed data set to show the power of the different types of change point analysis, so lets look at a real life example. Going back to our heart beat data which using the AMOC method gave the point 44 where it determined the mean of the data to change the most drastically. However, it would be expect that someone's heart beat would change fairly drastically when the person goes from resting to working out. A doctor might be interested in monitoring the patient to see if their heart rate is changing drastically when working out to see what is stressing the heart the most. In this example it would be useful to use a multiple change point method in order to find where the heart rate is changing the most. Using the code from the simulation data above but changing the data source produces the follow change points displayed in Figure (8).



(a) PELT

(b) Binary Segmentation

Figure 8: Heart Rate Change Points

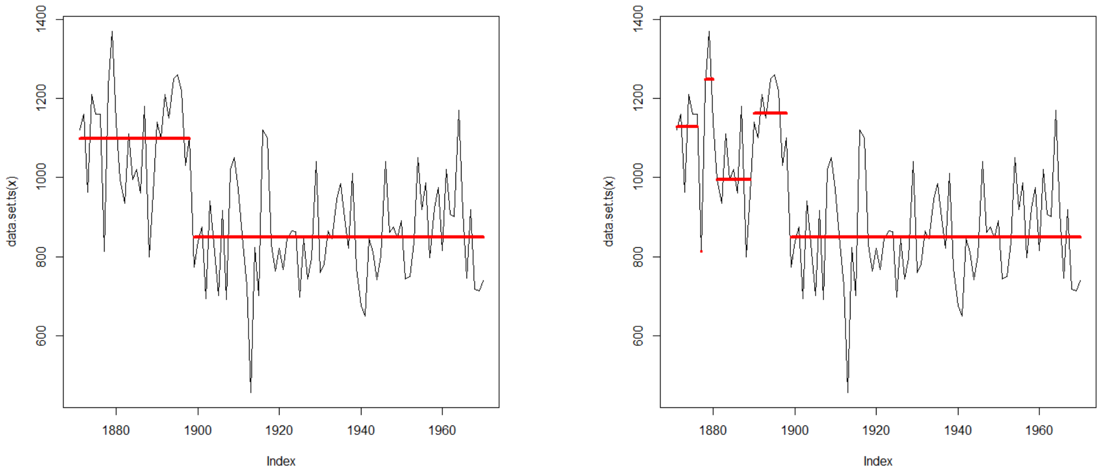
PELT method produces 100 different change points while the Binary Segmentation method produces 5 different points of change in mean. That means that there are at least 5 change points in our data set since both methods have at least 5 points. There are 1,160 data points in the heart rate data set so the 5 points that Binary Segmentation produced automatically might not be enough for the analysis. Luckily in the *cpt.mean* function when using Binary Segmentation method, Q is an input that determines the maximum number of change points. On the contrary it could be believed that the PELT method is over segmenting the data [2] and that is when a penalty would be used in order to consolidated the segments. PELT is an exact search function and under the default penalty (MBIC) determines that 100 points are optimal change points in the mean. Binary Segmentation is not an exact search so the results will not be as precise thus there are not as many change points, but the same default penalty was used for both methods. Either way these methods determined that there was still large changes in the mean beyond just one point.

5.2 Nile River

The final data set that will be looked at for change point analysis is volume of water flow for the Nile River. This data set consists of a 100 different points for the volume of water flow in the Nile River at Aswan for 100 years. This means the data is a time series which is shown in Figure (2).

To show the real benefit of using a multiple change point method instead a single point, Binary Segmentation and AMOC method are ran for the Nile River data. A single point of change in the data set might not be able to tell the whole story of the data which will be demonstrated in Figure (9).

```
> cpts(river.amoc)
[1] 28
```



(a) AMOC

(b) Binary Segmentation

Figure 9: Nile River Change Points

```
> cpts(river.bin)
[1]  6  7 10 19 28
```

It is interesting to see that all of the change points are within the first half of the time series data. It is apparent that at point 28, which is the year 1898, there was a big change in the mean of the time series. But what about the four other additional points before that year? Let's look at the first two data points produced by Binary Segmentation which are the years 1875 and 1876. This segment is super small and is over looked for the AMOC method, but it turns out that in 1876 and 1877 there was abnormally low flooding of the Nile that year which caused food shortages and starvation in Egypt. Had AMOC been the only method to analyze the data then I would have had no reason to look at the historical data for the Nile River in 1876. It shows that one point of change in a data set might not be sufficient enough to make conclusions about that data set.

5.3 Conclusion

Change point analysis gives analysts the ability to find multiple points of change in time series and sequence data. With this ability a better understanding of the data is now achievable by locating the points of change in the mean. It has been demonstrated the benefit to using multiple change point search method over just finding one change in the mean. There are many real world applications to the use of change point analysis which some have been addressed in this paper. As these methods become more research and improved upon, it will make such methods more efficient to run on large scale data sets. One thing to remember is that changes in mean were the only focus of this research which still leaves the analysis of changes in variance. It would be interesting to see how well change point analysis does when looking for changing in variances, which I plan to do future research about. Another area that would benefit from further testing is determining what the best penalty value for each method for the different data sets is and how it affects the results.

There are many applications for change point analysis that would prove to be very helpful that weren't covered in this paper. One area that change point analysis could prove to be helpful is the stock market. The stock market has many changes over time so finding just one point of change would not be the best way to analyze it. With change point analysis it would give economists the ability to see many changes in mean or variance of that stock. This would give them more insight into when is the right time to buy and sell certain stocks with more certainty. Change point analysis will prove to be extremely helpful as the world continues to get more into big data.

6 Code Appendix

```
if(!require(changepoint)){
  install.packages('changepoint')
}
library(changepoint) # required package


## Simulated Data
set.seed(10)
sim.data <- c(rnorm(100, 0, 1), rnorm(100, 1, 1), rnorm(100, 0, 1), rnorm(100, 0.2, 1))
dev.new()
ts.plot(sim.data, xlab = "Index", main = "Simulated Time Series Data")  # plot of the simulated data

sim.amoc <- cpt.mean(sim.data, method = "AMOC")
dev.new()
plot(sim.amoc, type = "l", xlab = "Index", cpt.width = 4)
cpts(sim.amoc)

sim.pelt <- cpt.mean(sim.data, method = "PELT")
dev.new()
plot(sim.pelt, type = "l", cpt.col = "blue", xlab = "Index", cpt.width = 4)
cpts(sim.pelt)  # ordered list of optimal number of changepoints

sim.binseg <- cpt.mean(sim.data, method = "BinSeg")
dev.new()
plot(sim.binseg, type = "l", xlab = "Index", cpt.width = 4)
cpts(sim.binseg)

system.time(cpt.mean(sim.data, method = "BinSeg"))
system.time(cpt.mean(sim.data, method = "PELT"))

dev.new() # Histogram of the cpts from differernt seed values
histo <- hist(c(79,97,79,192,192,90,90,207,197,99,99,202,202,100,98,200,200,100,200),
              main = "Histogram of Change Points",
              xlab = "Change Points")
```

```
## Heartbeat Data
load("D:/School/College - Senior/Thesis/HeartRate.RData")

dev.new()
ts.plot(HeartRate, main = "Heartbeat Data")

m.amoc <- cpt.mean(HeartRate, method = "AMOC")
dev.new()
plot(m.amoc, type = "l", xlab = "Index", cpt.width = 4)
cpts(m.amoc)

m.pelt <- cpt.mean(HeartRate, method = "PELT")
dev.new()
plot(m.pelt, type = "l", cpt.col = "blue", xlab = "Index", cpt.width = 4)
cpts(m.pelt)

m.binseg <- cpt.mean(HeartRate, method = "BinSeg", Q = 10)  # Q: maxium number of changepoint
dev.new()
plot(m.binseg, type = "l", xlab = "Index", cpt.width = 4)
cpts(m.binseg)

## Nile River annual flow at Aswan from 1871 to 1970
data(Nile)
dev.new()
ts.plot(Nile)

river.bin <- cpt.mean(Nile, method = "BinSeg")
dev.new()
plot(river.bin, type = "l", xlab = "Index", cpt.width = 4)
cpts(river.bin)

river.pelt <- cpt.mean(Nile,method = "PELT")
dev.new()
plot(river.pelt, type = "l", xlab = "Index", cpt.width = 4)
cpts(river.pelt)

river.amoc <- cpt.mean(Nile, method = "AMOC", Q = 10)
dev.new()
plot(river.amoc, type = "l", xlab = "Index", cpt.width = 4)
cpts(river.amoc)
```

References

[1] Jones, Jim *Egypt and Europe in the 19th Century.*
`http://courses.wcupa.edu/Jones/his312/lectures/egypt.html`

[2] Killick, Rebecca and Eckley, Idris *changepoint: An R package for Changepoint Analysis.* Journal of Statistical Software.

[3] Killick, Rebecca and Eckley, Idris and Fearnhead, P *Optimal detection of changepoint with a linear computational cost.* Journal of the American Statistical Association.

[4] Killick, Rebecca, *Introduction to optimal changepoint detection algorithms.*
`http://members.cbio.mines-paristech.fr/ thocking/change-tutorial/RK-CptWorkshop.html`

[5] Wambui, Gachomo; Waititu, Gichuhi; Wanjoya, Anthony *The Power of the Pruned Exact Linear Time (PELT) Test in Multiple Changepoint Detection.* American Journal of Theoretical and Applied Statistics

[6] Brad Jackson, Jeffrey D. Scargle, David Barnes, Sundararajan Arabhi, Alina Alt, Peter Gioumousis, Elyus Gwin,Paungkaew Sangtrakulcharoen, Linda Tan, and Tun Tao Tsai *An Algorithm for Optimal Partitioning if Data on an Interval.*
`https://arxiv.org/pdf/math/0309285.pdf`

[7] Maidstone, Robert; Hocking, Toby; Rigaill, Guillem; Fearnhead, Paul *On Optimal Multiple Changepoint Algorithms for Large Data.*
`https://arxiv.org/pdf/1409.1842.pdf`

[8] Killick, Rebecca *Package 'changepoint'.*
`https://github.com/rkillick/changepoint/`