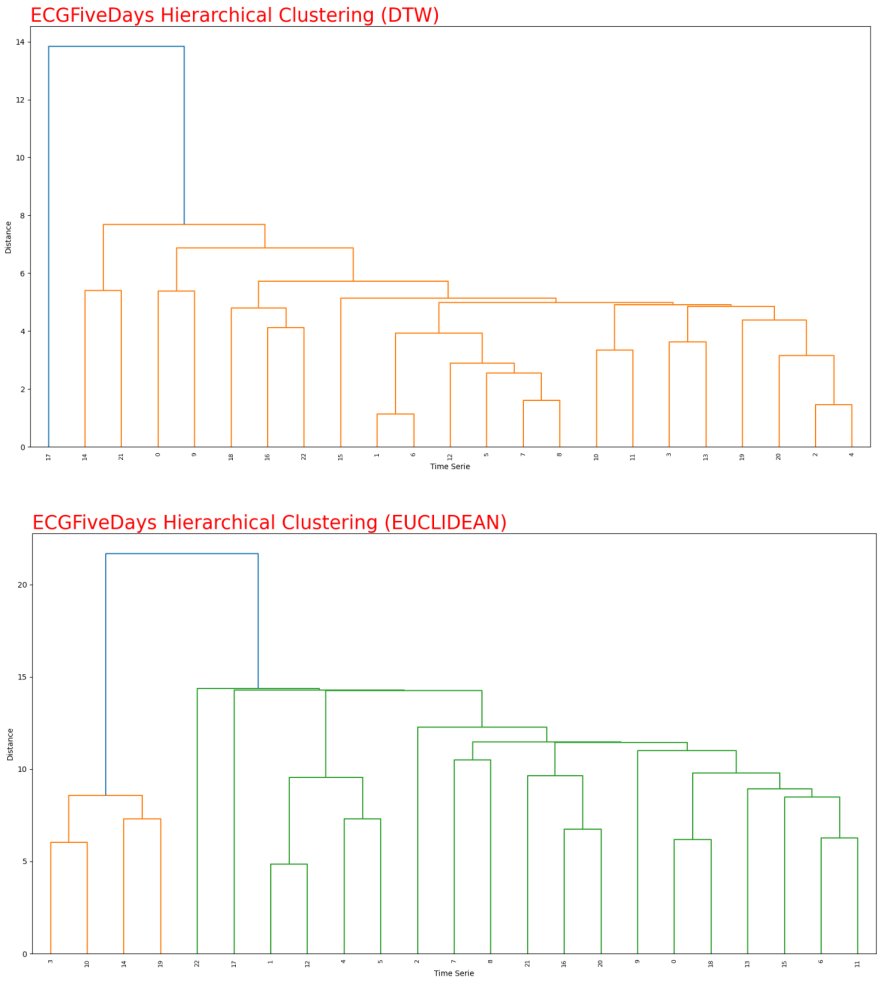
# Rapport TIW9 TP2
# Clustering de time-series

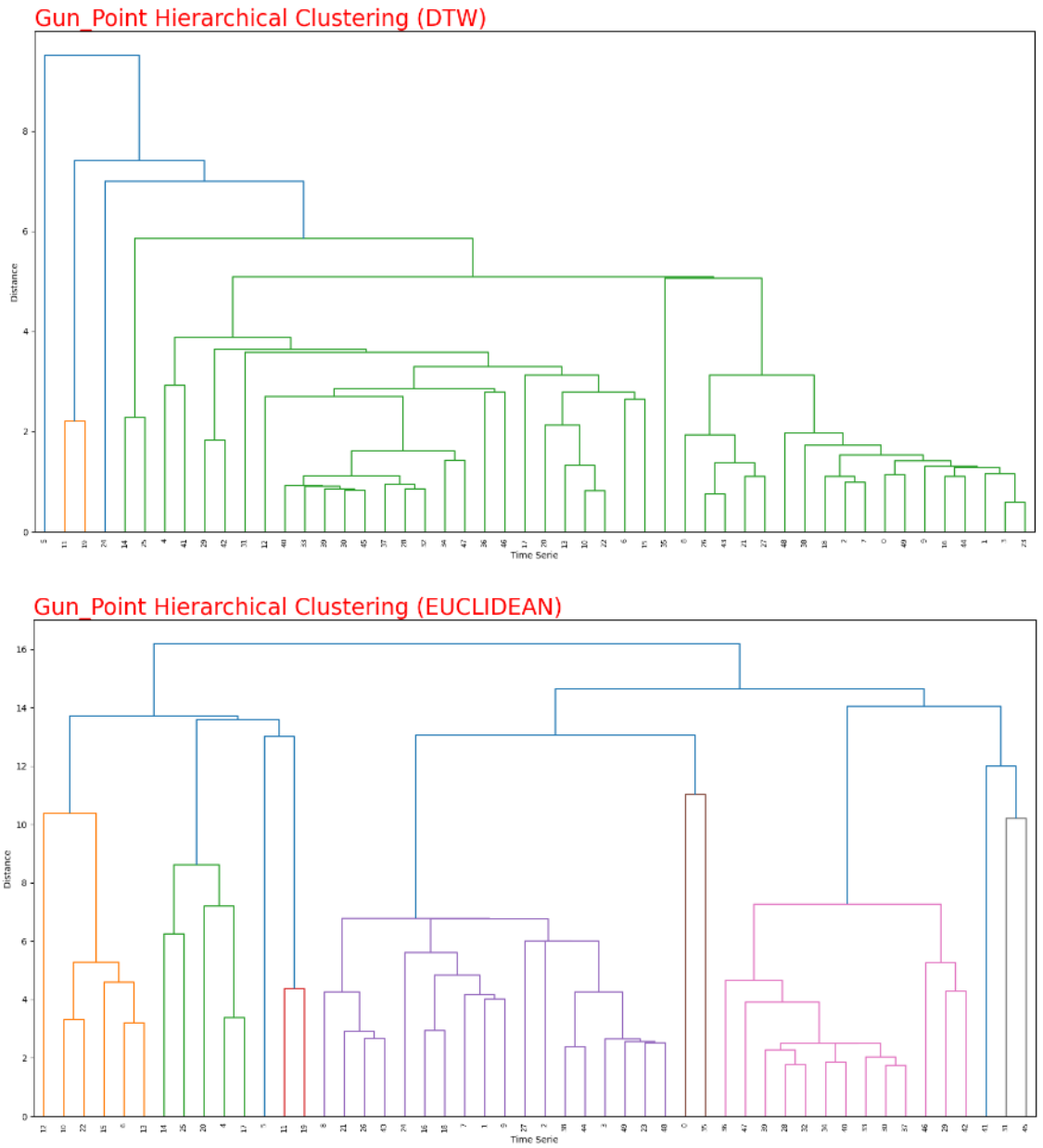
Binôme : Jeremy Thomas (11702137) et Julien Giraud (11704709)
Jeux de données : ECGFiveDays et Gun_Point

## Clustering hiérarchique avec DTW et Distance Euclidienne




*Clustering hiérarchique (DTW et Distance Euclidienne pour le jeu de données ECGFiveDays)*

Peu importe la méthode de calcul des distances utilisée, le cluster principal contient l'essentiel des données. Le reste, que nous pensons être du bruit, semble mieux géré par la méthode DTW. En effet, la méthode des distances Euclidiennes semble prendre en compte le bruit comme n'importe quelle autre donnée et l'utilise pour générer des clusters. Par ailleurs, en prenant en compte l'échelle des ordonnées, DTW semble plus précis car les distances entre les points sont presque deux fois plus faibles.



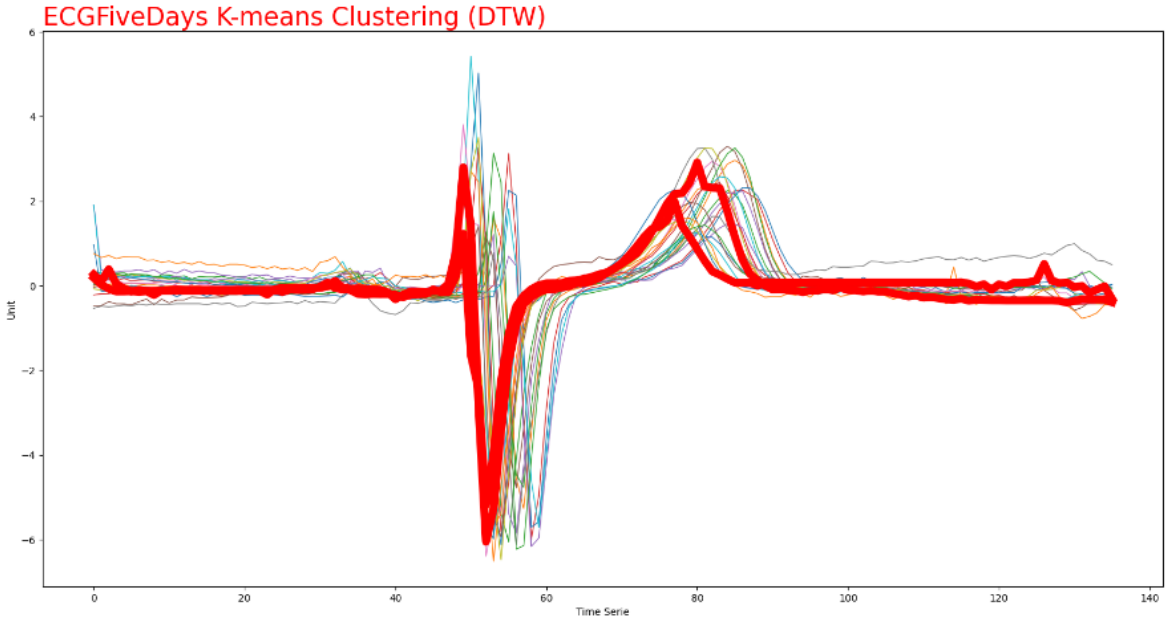

*Clustering hiérarchique (DTW et Distance Euclidienne pour le jeu de données Gun_Point)*

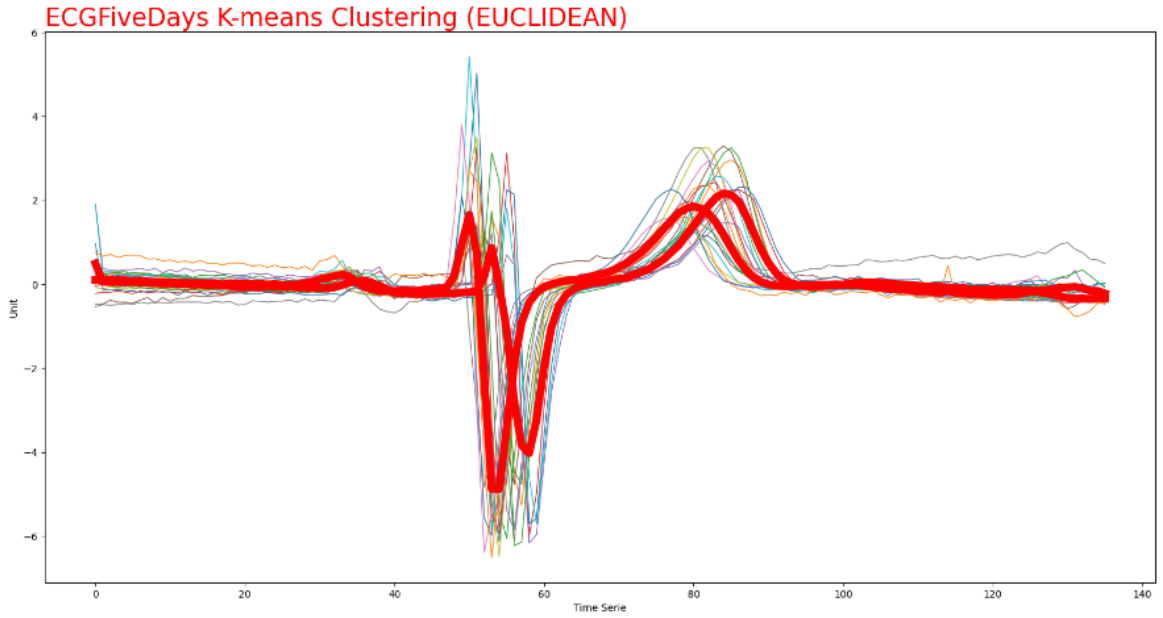
DTW rassemble une nouvelle fois l'essentiel des données dans le même cluster, le reste semble être du bruit.

Avec les distances euclidiennes, plusieurs clusters émergent mais ne semblent pas avoir de sens. Nous pensons que ces clusters sont dus au bruit qui interfère avec l'algorithme de clustering.

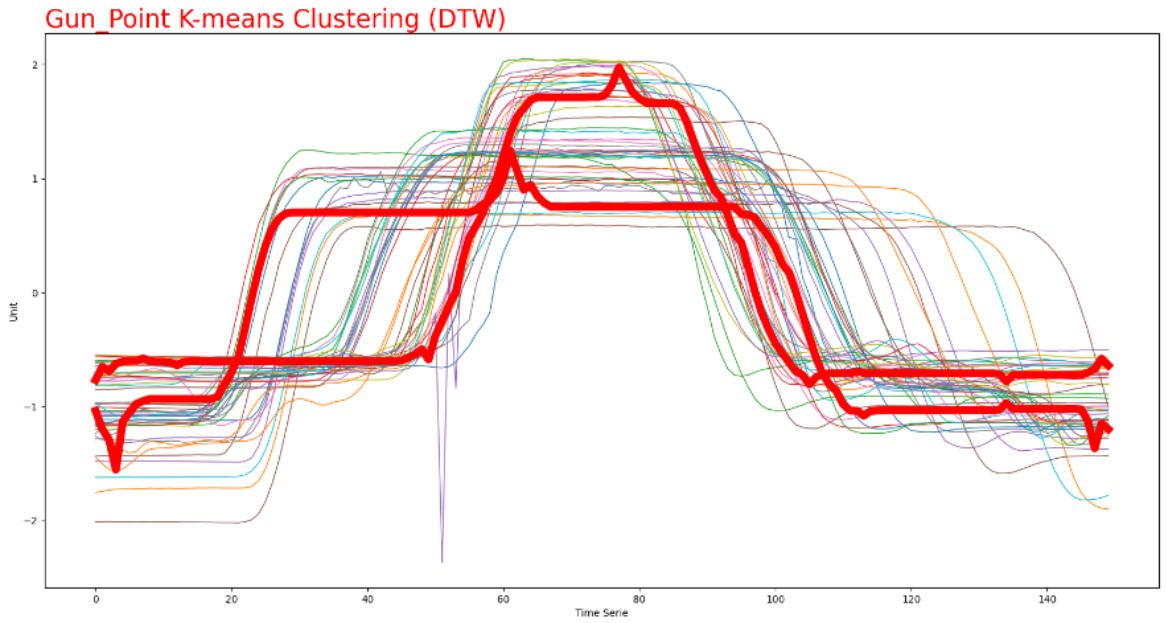
Encore une fois, les distances sont environ deux fois plus faibles avec DTW. Cette méthode de calcul semble permettre de détecter plus efficacement le bruit lors du clustering et donc de ne pas créer de clusters superflus.

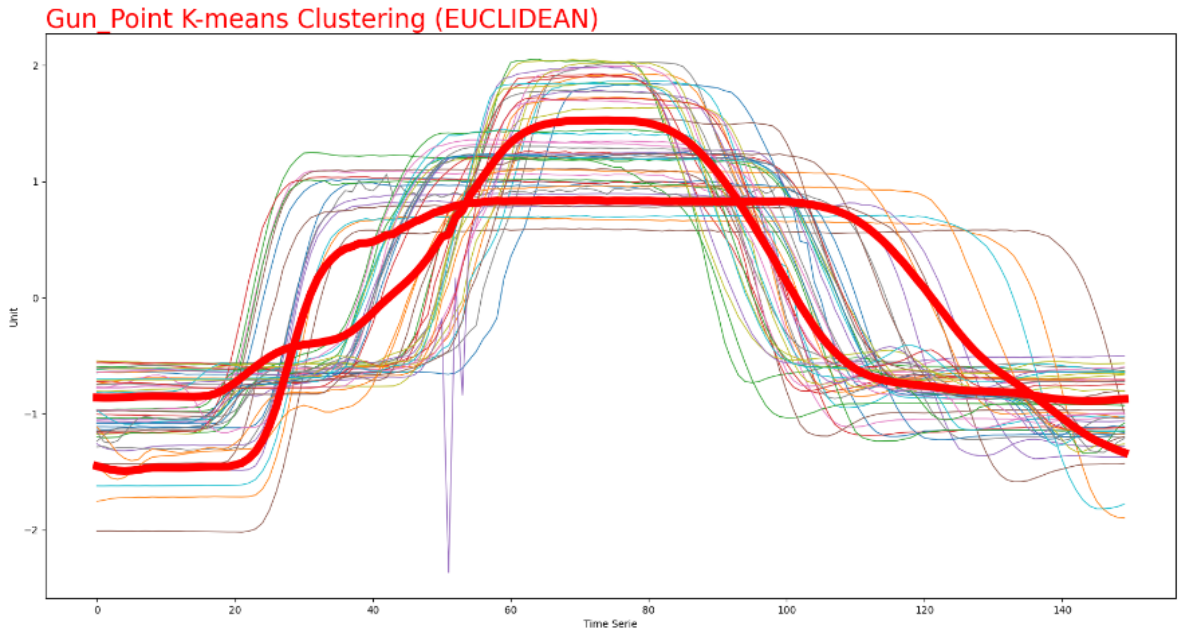
# Clustering K-means avec DTW et Distance Euclidienne




Dans l'utilisation des deux méthodes de calcul des distances, les deux clusters représentés par deux courbes rouge, représentent les deux classes du jeu de données. Elles suivent toutes les deux la tendance des courbes représentant les time-series. Avec prise en compte des distances Euclidiennes, la courbe est lissée.




Avec le jeu de données Gun_Point, la méthode DTW semble sensible aux variations de chaque time-series, alors que la courbe Euclidienne est lissée et suit la tendance moyenne des courbes sans être sensible aux variations de chacune.

# Comparaison des algorithmes de clustering : Clustering Hiérarchique et Clustering K-Means

## Mode de calcul

K-Means nécessite de connaître à l'avance le nombre de clusters à générer et décide aléatoirement du centre de ces derniers. Cela implique que deux exécutions de l'algorithme peuvent ne pas donner le même résultat. L'avantage du clustering hiérarchique est alors qu'il ne nécessite pas de connaître le jeu de données à l'avance.
En revanche, le clustering hiérarchique nécessite l'utilisation d'une matrice de n*n time-series, ce qui peut être particulièrement coûteux sur des time-series de grande taille. K-Means quant à lui peut travailler sur des séries de tailles différentes.

## Durée d'exécution du clustering

| Algorithme/Méthode de calcul des distances | DTW | Distance Euclidienne |
|---|---|---|
| K-Means | 3,5s (ECGFiveDays)<br>4,76s (Gun_Point) | 1,25s (ECGFiveDays)<br>2,23s (Gun_Point) |
| Clustering hiérarchique | 6,74s (ECGFiveDays)<br>37,5s (Gun_Point) | 0,54s (ECGFiveDays)<br>0,75s (Gun_Point) |

Intéressons nous aux durées d'exécution par rapport aux algorithmes de clustering.

Par la méthode DTW, l'algorithme K-means est toujours plus rapide que le clustering hiérarchique. Il est important de noter que la durée d'exécution n'est pas proportionnelle au volume de données. La durée d'exécution du clustering hiérarchique explose par rapport à K-Means lorsque le volume de données augmente : de 3.5s à 4.76 (+36%) pour K-Means et de 6.74 à 37.5s (+456%) pour le clustering hiérarchique. Cela peut s'expliquer par le fait que l'algorithme K-Means a une complexité quadratique $O(n^2)$ alors que le clustering hiérarchique une complexité cubique $O(n^3)$.

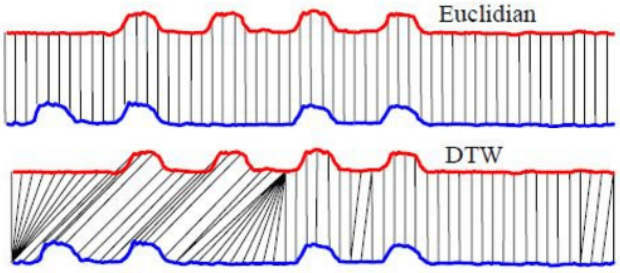
Par la méthode des distances euclidiennes, le clustering hiérarchique est plus rapide. Nous n'avons à l'heure actuelle pas trouvé d'explication à cela.

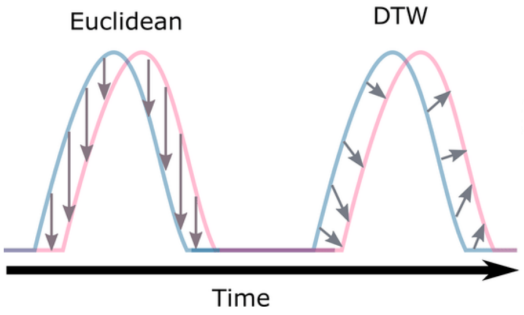
# Comparaison des méthodes de calcul des distances : DTW et Euclidienne

## Mode de calcul

L'algorithme DTW (Dynamic Time Warping ou Déformation Temporelle Dynamique) est un algorithme qui permet de calculer la similarité entre deux time series. DTW prend en compte les décalages temporels et permet la comparaison d'un point avec un ou plusieurs autres points.
La méthode de calcul des distances Euclidiennes quant à elle ne mesure que la distance de point à point et de manière linéaire.

Le calcul des similarités point à point de manière linéaire est beaucoup plus rapide que point à plusieurs en prenant en compte le décalage temporel.
Cela implique que l'algorithme DTW est plus précis mais plus long à s'exécuter. Le calcul par la distance Euclidienne est moins précis mais beaucoup plus rapide.




## Précision

Pour déterminer quelle méthode de calcul des distances (DTW ou Euclidienne) est la plus précise pour le clustering K-Means, nous avons utilisé la fonction tslearn.metrics.dtw qui donne un score de similarité entre deux time-series tel que:

$$Score(S1, S2) = \sqrt{\sum_{(i,j)}^{n} \|S1i - S2j\|^2}$$

([https://tslearn.readthedocs.io/en/stable/gen_modules/metrics/tslearn.metrics.dtw.html](https://tslearn.readthedocs.io/en/stable/gen_modules/metrics/tslearn.metrics.dtw.html))

Plus les time-series sont similaires, plus le score est faible (Lower is better). Si le score vaut 0, les deux time-series sont identiques. Ainsi, en considérant le cluster généré comme une time-serie, nous avons calculé le score moyen pour les deux méthodes de calcul des distances par la formule suivante:

$$\frac{\sum_{i}^{n} ScoreMinimum(TimeSerie(i), Cluster)}{n}$$

(avec n le nombre de série)

Les résultats montrent que la similarité entre les time-series et le cluster généré est plus élevée par la méthode DTW qu'avec l'utilisation des distances Euclidiennes.

Avec le jeu de données ECGFiveDays, nous avons estimé sur plusieurs exécutions une similarité avec DTW environ 50% plus élevée qu'avec les distances Euclidiennes:

```
Score moyen entre les time-series et le cluster généré (DTW): 2.15 (Lower is better)
Score moyen entre les time-series et le cluster généré (distance Euclidiennes): 3.13 (Lower is better)
```

Avec le jeu de données Gun_Point, nous avons estimé sur plusieurs exécutions une similarité avec DTW environ 100% plus élevée, soit deux fois plus élevée, qu'avec les distances Euclidiennes :

```
Score moyen entre les time-series et le cluster généré (DTW): 0.95 (Lower is better)
Score moyen entre les time-series et le cluster généré (distance Euclidiennes): 1.89 (Lower is better)
```

Nous en déduisons que le clustering K-Means utilisant DTW génère des clusters qui représentent mieux le jeu de données.

## Durée d'exécution

Une manière simple de comparer deux méthodes de calcul est de chronométrer leur temps d'exécution.

| Algorithme/Méthode de calcul | DTW | Distance Euclidienne |
|---|---|---|
| K-Means | 3,5s (ECGFiveDays)<br>4,76s (Gun_Point) | 1,25s (ECGFiveDays)<br>2,23s (Gun_Point) |
| Clustering hiérarchique | 6,74s (ECGFiveDays)<br>37,5s (Gun_Point) | 0,54s (ECGFiveDays)<br>0,75s (Gun_Point) |

Intéressons nous aux durées d'exécution par rapport aux méthodes de calcul des distances.

On s'aperçoit que peu importe l'algorithme de clustering utilisé, la méthode avec calcul des distances Euclidiennes est toujours plus rapide que DTW. Cela se justifie par le mode de calcul de chacune des deux méthodes : point à point pour la distance Euclidienne et point à plusieurs pour DTW. DTW nécessite de calculer la matrice des distances entre chaque série de données.

# Conclusion

Pour conclure, nous avons étudié deux algorithmes de clustering ainsi que deux méthodes de calcul des distances. Par l'exécution des algorithmes et des méthodes sur deux jeux de données de taille différente, nous avons remarqué que DTW était plus lent mais également plus précis que les distances euclidiennes. Le clustering hiérarchique donne plus d'informations sur la similarité entre les time-series que K-Means mais sa complexité fait que sur de gros volume de données la méthode DTW peut ne pas être adaptée à cet algorithme.