Efficient Multivariate Analysis of Change Points

The e-cp3o procedure

Nicholas A. James
Operations Research
https://courses.cit.cornell.edu/nj89

David S. Matteson
Statistical Sciences
https://courses.cit.cornell.edu/dm484

May 30, 2015

Change Point Analysis

- Partition time series into homogeneous segments.

Change Point Analysis

- Partition time series into homogeneous segments.
- Also refereed to by other names.

Change Point Analysis

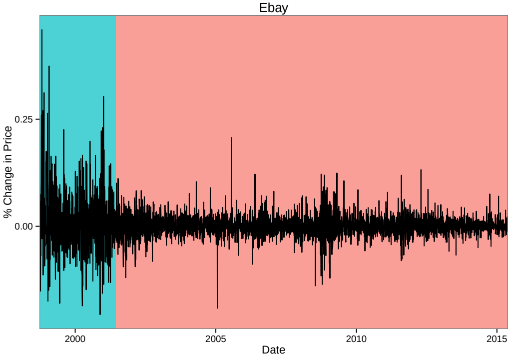
- Partition time series into homogeneous segments.
- Also refereed to by other names.
  - Break points
  - Breakouts
  - Regime changes

Change Point Analysis

- Partition time series into homogeneous segments.
- Also refereed to by other names.
    - Break points
    - Breakouts
    - Regime changes
- Change points exist in real data.

Change Point Analysis

- Partition time series into homogeneous segments.
- Also refereed to by other names.
  - Break points
  - Breakouts
  - Regime changes
- Change points exist in real data.



Difficulties

- Distribution of observations is unknown.

Difficulties

- Distribution of observations is unknown.
- Number of change points is unknown.

Difficulties

- Distribution of observations is unknown.
- Number of change points is unknown.
- Unknown distance between change points.

Difficulties

- Distribution of observations is unknown.
- Number of change points is unknown.
- Unknown distance between change points.
- Difficult even if number of change points is known.

Difficulties

- Distribution of observations is unknown.
- Number of change points is unknown.
- Unknown distance between change points.
- Difficult even if number of change points is known.
  - Exponential in number of change points.

Divergence Measure

- Let X_1 and Y_1 be independent random vectors. And (X_2, Y_2) and iid copy of (X_1, Y_2).

Divergence Measure

- Let X_1 and Y_1 be independent random vectors. And (X_2, Y_2) and iid copy of (X_1, Y_2).

- The energy distance between the distributions of X_1 and Y_1

Divergence Measure

- Let X_1 and Y_1 be independent random vectors. And (X_2, Y_2) and iid copy of (X_1, Y_2).

- The energy distance between the distributions of X_1 and Y_1

$$\mathcal{E}(X_1, Y_1|\alpha) = 2E|X_1 - Y_1|^\alpha - E|X_1 - X_2|^\alpha - E|Y_1 - Y_2|^\alpha$$

$$= 2 \cdot \texttt{between} - \texttt{X within} - \texttt{Y within}$$

Divergence Measure

Let $\mathbf{X}_n = \{x_1, x_2, \ldots, x_n\}$ and $\mathbf{Y}_m = \{y_1, y_2, \ldots, y_m\}$.

Divergence Measure

Let $\mathbf{X}_n = \{x_1, x_2, \ldots, x_n\}$ and $\mathbf{Y}_m = \{y_1, y_2, \ldots, y_m\}$.

Their sample divergence is given by

$$\begin{aligned}\widehat{\mathcal{E}}(\mathbf{X}_n, \mathbf{Y}_m | \alpha) = & \frac{2}{mn} \sum_{i,j} |x_i - y_j|^{\alpha} \\ & - \binom{n}{2}^{-1} \sum_{i<j} |x_i - x_j|^{\alpha} \\ & - \binom{m}{2}^{-1} \sum_{i<j} |y_i - y_j|^{\alpha}.\end{aligned}$$

Finding Change Points

Let $k > 0$ and

$$\widehat{\mathcal{R}}(\mathbf{X}_n, \mathbf{Y}_m | \alpha) = \frac{mn}{(m+n)^2}\, \widehat{\mathcal{E}}(\mathbf{X}_n, \mathbf{Y}_m | \alpha).$$

Finding Change Points

Let $k > 0$ and

$$\widehat{\mathcal{R}}(\mathbf{X}_n, \mathbf{Y}_m|\alpha) = \frac{mn}{(m+n)^2}\ \widehat{\mathcal{E}}(\mathbf{X}_n, \mathbf{Y}_m|\alpha).$$

The best partitioning of time series $Z_1, Z_2, \ldots, Z_T \in \mathbb{R}^d$, with k change points is given

$$\beta_k(T) = \max_{\tau_1,\tau_2,\ldots,\tau_k} \widehat{\mathcal{R}}(C_0, C_1|\alpha) + \widehat{\mathcal{R}}(C_1, C_2|\alpha) + \cdots + \widehat{\mathcal{R}}(C_{k-1}, C_k|\alpha).$$

Finding Change Points

Let $k > 0$ and

$$\widehat{\mathcal{R}}(\mathbf{X}_n, \mathbf{Y}_m|\alpha) = \frac{mn}{(m+n)^2}\,\widehat{\mathcal{E}}(\mathbf{X}_n, \mathbf{Y}_m|\alpha).$$

The best partitioning of time series $Z_1, Z_2, \ldots, Z_T \in \mathbb{R}^d$, with k change points is given

$$\beta_k(T) = \max_{\tau_1,\tau_2,\ldots,\tau_k} \widehat{\mathcal{R}}(C_0, C_1|\alpha) + \widehat{\mathcal{R}}(C_1, C_2|\alpha) + \cdots + \widehat{\mathcal{R}}(C_{k-1}, C_k|\alpha).$$

With $C_i = \{Z_{\tau_{i-1}+1}, Z_{\tau_{i-1}+2}, \ldots, Z_{\tau_i}\}$

Finding Change Points

Let $k > 0$ and

$$\widehat{\mathcal{R}}(\mathbf{X}_n, \mathbf{Y}_m|\alpha) = \frac{mn}{(m+n)^2}\,\widehat{\mathcal{E}}(\mathbf{X}_n, \mathbf{Y}_m|\alpha).$$

The best partitioning of time series $Z_1, Z_2, \ldots, Z_T \in \mathbb{R}^d$, with k change points is given

$$\beta_k(T) = \max_{\tau_1,\tau_2,\ldots,\tau_k} \widehat{\mathcal{R}}(C_0, C_1|\alpha)+\widehat{\mathcal{R}}(C_1, C_2|\alpha)+\cdots+\widehat{\mathcal{R}}(C_{k-1}, C_k|\alpha).$$

With $C_i = \{Z_{\tau_{i-1}+1}, Z_{\tau_{i-1}+2}, \ldots, Z_{\tau_i}\}$

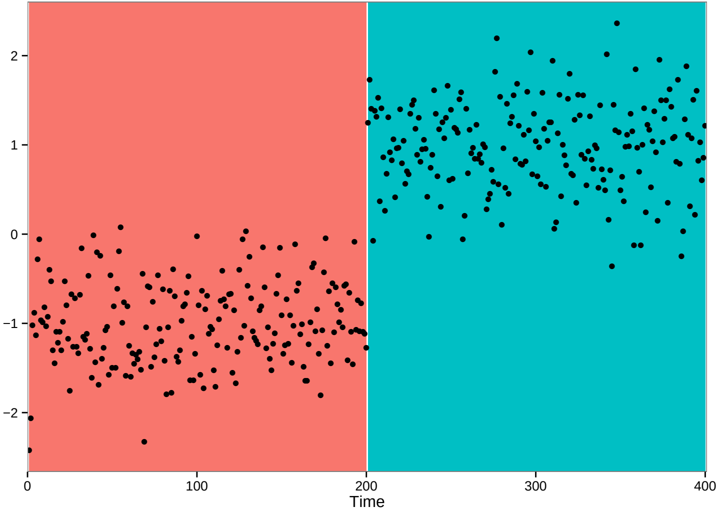
Can find $\hat{\tau}_1, \ldots, \hat{\tau}_k$ in $\mathcal{O}(kT^3)$ time.

Finding Change Points

Finding change points using $\widehat{\mathcal{R}}$ is too slow.

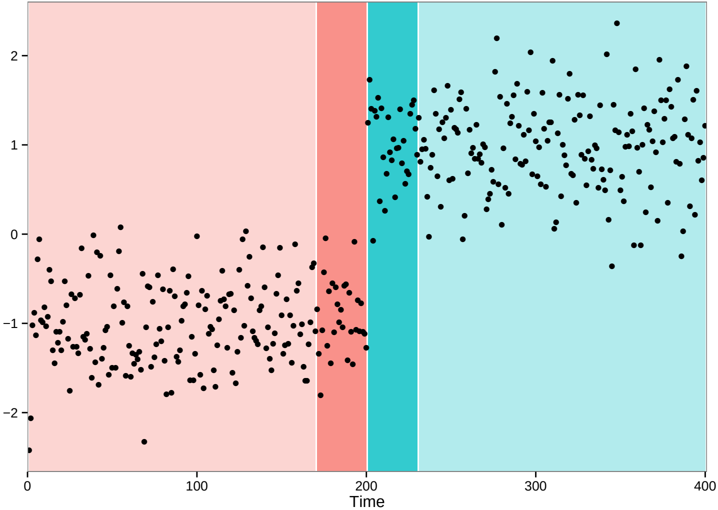
Finding Change Points

Finding change points using $\widehat{\mathcal{R}}$ is too slow.



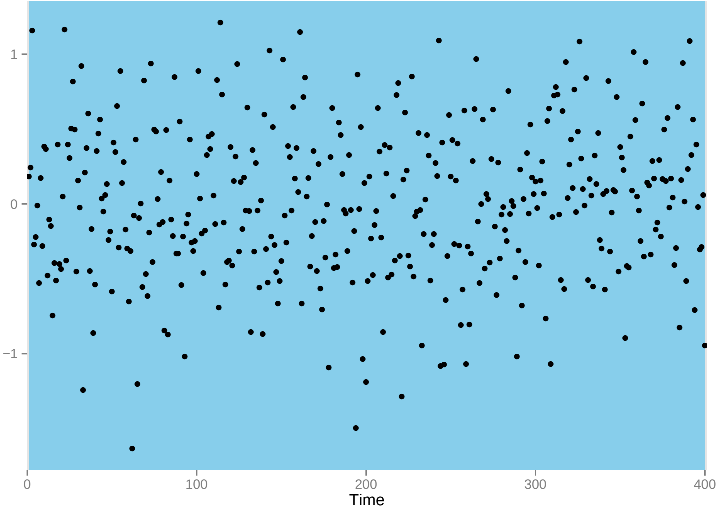
Finding Change Points

Finding change points using $\widehat{\mathcal{R}}$ is too slow.



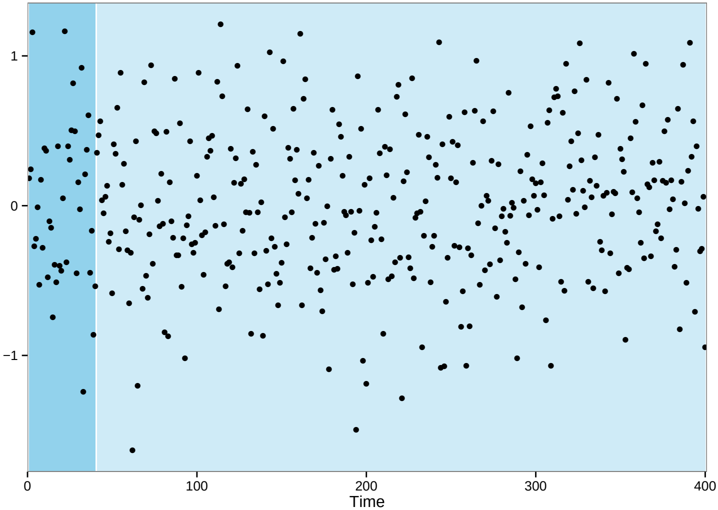
Finding Change Points

Finding change points using $\widehat{\mathcal{R}}$ is too slow.



Finding Change Points

Finding change points using $\widehat{\mathcal{R}}$ is too slow.



Finding Change Points

Instead use

$$\widetilde{\mathcal{E}}(\mathbf{X}_n, \mathbf{Y}_m | \alpha, \delta) = \frac{2}{\# B^\delta} \sum_{(i,j)\in B^\delta} |x_i - y_j|^\alpha - \frac{1}{\# W_x^\delta} \sum_{(i,j)\in W_x^\delta} |x_i - x_j|^\alpha - \frac{1}{\# W_y^\delta} \sum_{(i,j)\in W_y^\delta} |y_i - y_j|^\alpha.$$

Finding Change Points

Instead use

$$\widetilde{\mathcal{E}}(\mathbf{X}_n, \mathbf{Y}_m | \alpha, \delta) = \frac{2}{\#B^\delta} \sum_{(i,j) \in B^\delta} |x_i - y_j|^\alpha - \frac{1}{\#W_x^\delta} \sum_{(i,j) \in W_x^\delta} |x_i - x_j|^\alpha - \frac{1}{\#W_y^\delta} \sum_{(i,j) \in W_y^\delta} |y_i - y_j|^\alpha.$$

Sets B^δ, W_x^δ, and W_y^δ are sets of index pairs.

Finding Change Points

Using $\widetilde{\mathcal{R}}$ allows us to find $\hat{\tau}_1, \ldots, \hat{\tau}_k$ in $\mathcal{O}(kT^2)$ time.

Finding Change Points

Using $\widetilde{\mathcal{R}}$ allows us to find $\hat{\tau}_1, \ldots, \hat{\tau}_k$ in $\mathcal{O}(kT^2)$ time.

Still one significant negative property of this approach:

Finding Change Points

Using $\widetilde{\mathcal{R}}$ allows us to find $\hat{\tau}_1, \ldots, \hat{\tau}_k$ in $\mathcal{O}(kT^2)$ time.

Still one significant negative property of this approach:

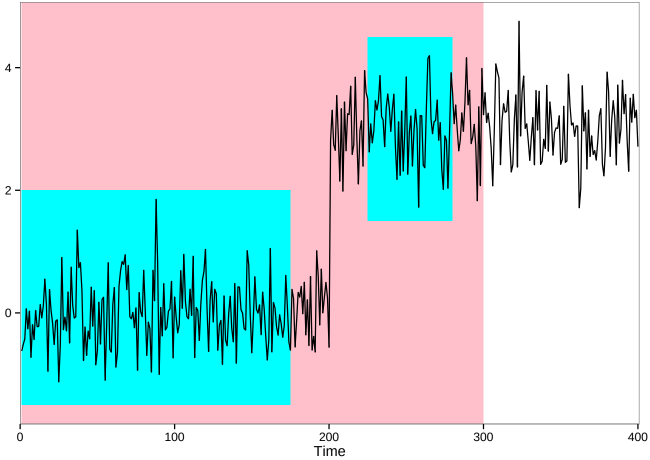
- performs many unnecessary calculations

Finding Change Points

Remove points from the search space that have probability less than ϵ of being a true change point.

Finding Change Points

Remove points from the search space that have probability less than ϵ of being a true change point.



Finding Change Points

Bound the change created by an additional change point.

Finding Change Points

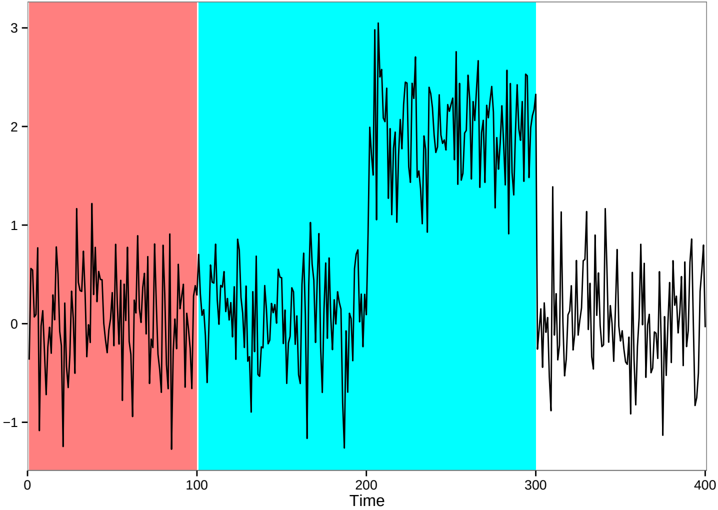
Bound the change created by an additional change point.

$$\widetilde{\mathcal{R}}(Z_{v+1}^{t}, Z_{t+1}^{u}|\alpha) - \widetilde{\mathcal{R}}(Z_{v+1}^{t}, Z_{t+1}^{s}|\alpha) - \widetilde{\mathcal{R}}(Z_{t+1}^{s}, Z_{s+1}^{u}|\alpha) < \Gamma$$

Finding Change Points

Bound the change created by an additional change point.

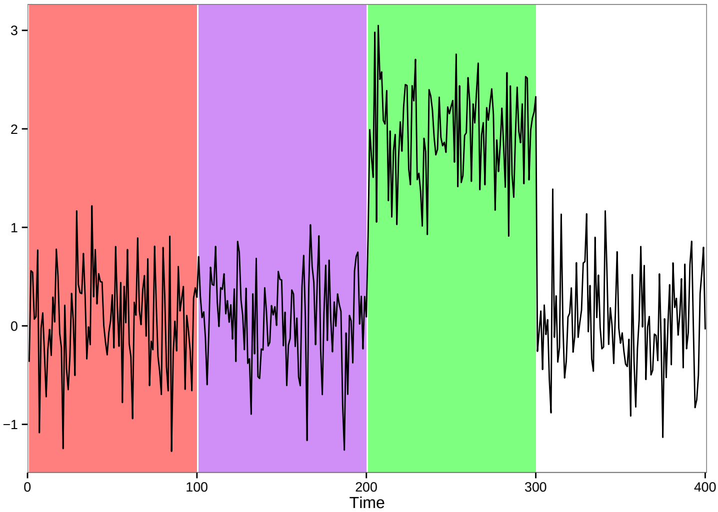
$$\widetilde{\mathcal{R}}(\mathbf{Z^t_{v+1}}, \mathbf{Z^u_{t+1}}|\alpha) - \widetilde{\mathcal{R}}(Z^t_{v+1}, Z^s_{t+1}|\alpha) - \widetilde{\mathcal{R}}(Z^s_{t+1}, Z^u_{s+1}|\alpha) < \Gamma$$



Finding Change Points

Bound the change created by an additional change point.

$$\widetilde{\mathcal{R}}(Z_{v+1}^{t}, Z_{t+1}^{u}|\alpha) - \widetilde{\mathcal{R}}(\mathbf{Z}_{\mathbf{v+1}}^{\mathbf{t}}, \mathbf{Z}_{\mathbf{t+1}}^{\mathbf{s}}|\alpha) - \widetilde{\mathcal{R}}(\mathbf{Z}_{\mathbf{t+1}}^{\mathbf{s}}, \mathbf{Z}_{\mathbf{s+1}}^{\mathbf{u}}|\alpha) < \Gamma$$



Finding Change Points

Unknown distributions make finding Γ impossible.

Finding Change Points

Unknown distributions make finding Γ impossible.

Consider a probabilistic version

$$\mathbb{P}\left(\widetilde{\mathcal{R}}(Z_{v+1}^{t}, Z_{t+1}^{u}|\alpha) - \widetilde{\mathcal{R}}(Z_{v+1}^{t}, Z_{t+1}^{s}|\alpha) - \widetilde{\mathcal{R}}(Z_{t+1}^{s}, Z_{s+1}^{u}|\alpha) \geq \Gamma_{\epsilon}\right) \leq \epsilon$$

Finding Change Points

Unknown distributions make finding Γ impossible.

Consider a probabilistic version

$$\mathbb{P}\left(\widetilde{\mathcal{R}}(Z_{v+1}^{t}, Z_{t+1}^{u}|\alpha) - \widetilde{\mathcal{R}}(Z_{v+1}^{t}, Z_{t+1}^{s}|\alpha) - \widetilde{\mathcal{R}}(Z_{t+1}^{s}, Z_{s+1}^{u}|\alpha) \geq \Gamma_{\epsilon}\right) \leq \epsilon$$

Remove t from the search space if

$$\zeta_k(t) + \widetilde{\mathcal{R}}(Z_{v+1}^{t}, Z_{t+1}^{s}|\alpha) + \Gamma_{\epsilon} < \zeta_k(s)$$

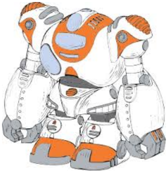
Using e-cp3o

The e-cp3o algorithm is currently available in the `ecp` package.

Using e-cp3o

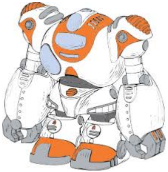
The e-cp3o algorithm is currently available in the **`ecp`** package.

```
e.cp3o(Z, K, delta, alpha, eps)
```



Using e-cp3o

The e-cp3o algorithm is currently available in the **`ecp`** package.

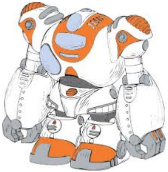


```
e.cp3o(Z, K, delta, alpha, eps)
```

- `Z`: Time series as a matrix

Using e-cp3o

The e-cp3o algorithm is currently available in the **ecp** package.



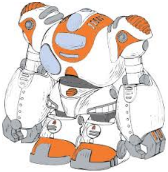
```
e.cp3o(Z, K, delta, alpha, eps)
```

- `Z`: Time series as a matrix
- `K`: Maximum number of change points to fit

Using e-cp3o

The e-cp3o algorithm is currently available in the **ecp** package.

```
e.cp3o(Z, K, delta, alpha, eps)
```

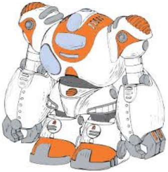


- `Z`: Time series as a matrix
- `K`: Maximum number of change points to fit
- `delta`: Minimum number of observations between change points

Using e-cp3o

The e-cp3o algorithm is currently available in the **ecp** package.

```
e.cp3o(Z, K, delta, alpha, eps)
```

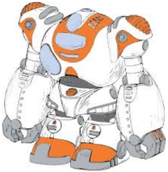


- `Z`: Time series as a matrix
- `K`: Maximum number of change points to fit
- `delta`: Minimum number of observations between change points
- `alpha`: Distance weighting

Using e-cp3o

The e-cp3o algorithm is currently available in the **ecp** package.

```
e.cp3o(Z, K, delta, alpha, eps)
```



- `Z`: Time series as a matrix
- `K`: Maximum number of change points to fit
- `delta`: Minimum number of observations between change points
- `alpha`: Distance weighting
- `eps`: Pruning probability

Gold Price (USD)

Look for change in the price of gold (USD) from January 1, 2000 to January 1, 2015. This results in 3789 observations.

Gold Price (USD)

Look for change in the price of gold (USD) from January 1, 2000 to January 1, 2015. This results in 3789 observations.

Obtain data using `Quandl`.

Gold Price (USD)

Look for change in the price of gold (USD) from January 1, 2000 to January 1, 2015. This results in 3789 observations.

Obtain data using **`Quandl`**.

```
> library("Quandl")
```

Gold Price (USD)

Look for change in the price of gold (USD) from January 1, 2000 to January 1, 2015. This results in 3789 observations.

Obtain data using **Quandl**.

```
> library("Quandl")

> gold = Quandl("BUNDESBANK/BBK01_WT5511",
                type="xts", transformation="rdiff",
                trim_start="1999-12-31",
                trim_end="2015-01-01")
```

Gold Price (USD)

Use **`quantmod`** to create time series plot.

Gold Price (USD)

Use **quantmod** to create time series plot.

```
> library("quantmod")
```

Gold Price (USD)

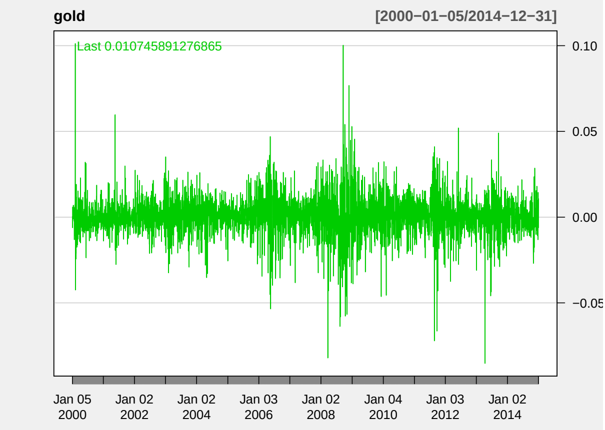
Use **quantmod** to create time series plot.

```
> library("quantmod")
> chartSeries(gold, theme=chartTheme("white"))
```

Gold Price (USD)

Use **quantmod** to create time series plot.

```
> library("quantmod")
> chartSeries(gold, theme=chartTheme("white"))
```



Gold Price (USD)

Use `ecp` package to find change points.

Gold Price (USD)

Use **ecp** package to find change points.

```
> library("ecp")
```

Gold Price (USD)

Use **ecp** package to find change points.

```
> library("ecp")
> res = e.cp3o(Z=gold, K=20, delta=6, alpha=1,
               eps=0.01)
```

Gold Price (USD)

Use **ecp** package to find change points.

```
> library("ecp")
> res = e.cp3o(Z=gold, K=20, delta=6, alpha=1,
               eps=0.01)
> res$time
```

Gold Price (USD)

Use **ecp** package to find change points.

```
> library("ecp")
> res = e.cp3o(Z=gold, K=20, delta=6, alpha=1,
               eps=0.01)
> res$time
 [1] 10.866
```

Gold Price (USD)

Use **ecp** package to find change points.

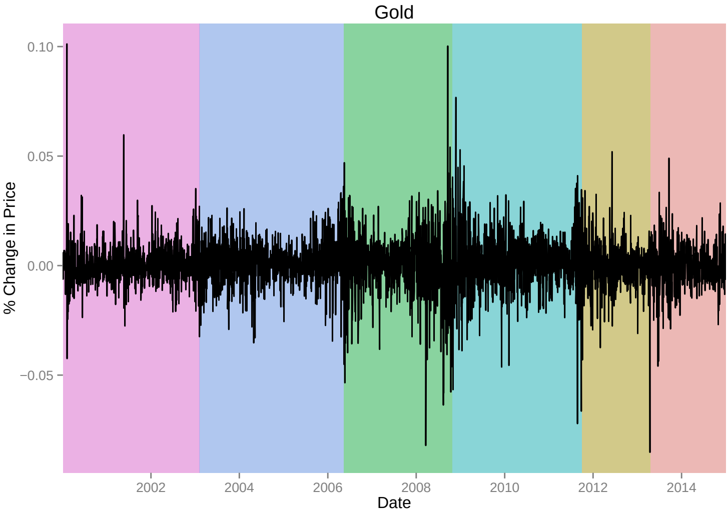
```
> library("ecp")
> res = e.cp3o(Z=gold, K=20, delta=6, alpha=1,
                  eps=0.01)
> res$time
 [1] 10.866
> index(gold[res$estimates])
```

Gold Price (USD)

Use **ecp** package to find change points.

```
> library("ecp")
> res = e.cp3o(Z=gold, K=20, delta=6, alpha=1,
               eps=0.01)
> res$time
 [1] 10.866
> index(gold[res$estimates])
 [1] "2003-02-05" "2006-05-12" "2008-10-24"
     "2011-09-29" "2013-04-17"
```

Gold Price (USD)



Amazon & Ebay

Look for changes returns for Amazon and Ebay stock, from January 1999 to January 2015. This time series has 4038 observations.

Amazon & Ebay

Look for changes returns for Amazon and Ebay stock, from January 1999 to January 2015. This time series has 4038 observations.

```
> ebay = Quandl("GOOG/NASDAQ_EBAY",
                transformation="rdiff",
                trim_start="1999-01-01",
                trim_end="2014-12-31")

> amazon = Quandl("GOOG/NASDAQ_AMZN",
                  transformation="rdiff",
                  trim_start="1999-01-01",
                  trim_end="2014-12-31")
```

Amazon & Ebay

Look for changes returns for Amazon and Ebay stock, from January 1999 to January 2015. This time series has 4038 observations.

```
> ebay = Quandl("GOOG/NASDAQ_EBAY",
                transformation="rdiff",
                trim_start="1999-01-01",
                trim_end="2014-12-31")

> amazon = Quandl("GOOG/NASDAQ_AMZN",
                  transformation="rdiff",
                  trim_start="1999-01-01",
                  trim_end="2014-12-31")

> prices = matrix(c(ebay$Close,
                  amazon$Close),ncol=2,byrow=T)
```

Amazon & Ebay

```
> res = e.cp3o(Z=prices, K=20, delta=6, alpha=1,
               eps=0.01)
```

Amazon & Ebay

```
> res = e.cp3o(Z=prices, K=20, delta=6, alpha=1,
               eps=0.01)
```

```
> res$time
```

Amazon & Ebay

```
> res = e.cp3o(Z=prices, K=20, delta=6, alpha=1,
                eps=0.01)

> res$time
 [1] 22.142
```

Amazon & Ebay

```
> res = e.cp3o(Z=prices, K=20, delta=6, alpha=1,
               eps=0.01)
```

```
> res$time
 [1] 22.142
```

```
> ebay$Date[res$estimates]
```

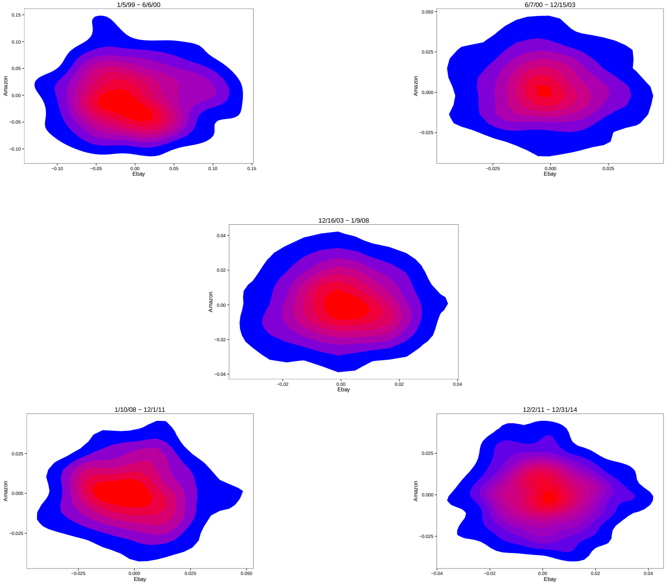
Amazon & Ebay

```
> res = e.cp3o(Z=prices, K=20, delta=6, alpha=1,
               eps=0.01)
```

```
> res$time
 [1] 22.142
```

```
> ebay$Date[res$estimates]
 [1] "2011-11-30" "2008-06-18" "2003-12-16"
     "1999-12-28"
```

Amazon & Ebay


1/5/99 – 6/6/00
Amazon
Ebay
6/7/00 – 12/15/03
Amazon
Ebay
12/16/03 – 1/9/08
Amazon
Ebay
1/10/08 – 12/1/11
Amazon
Ebay
12/2/11 – 12/31/14
Amazon
Ebay

Bibliography I

Bellman, R. (1952), "On the Theory of Dynamic Programming," *Proceedings of the National Academy of Sciences of the United States of America*, 38(8), 716.

Brodsky, E., and Darkhovsky, B. S. (1993), *Nonparametric Methods in Change Point Problems*, number 243 Springer.

Jackson, B., Scargle, J. D., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., Sangtrakulcharoen, P., Tan, L., and Tsai, T. T. (2005), "An algorithm for optimal partitioning of data on an interval," *Signal Processing Letters, IEEE*, 12(2), 105–108.

James, N. A., and Matteson, D. S. (2014), "ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data," *Journal of Statistical Software*, 62(7), 1–25.

James, N. A., and Matteson, D. S. (2015), "Change Points via Probabilistically Pruned Objectives," *arXiv preprint arXiv:1505.04302*, .

Killick, R., Fearnhead, P., and Eckley, I. (2012), "Optimal Detection of Changepoints With a Linear Computational Cost," *Journal of the American Statistical Association*, 107(500), 1590–1598.

Matteson, D. S., and James, N. A. (2014), "A Nonparametric Approach for Multiple Change Point Analysis of Multivariate Data," *Journal of the American Statistical Association*, 109(505), 334 – 345.

McTaggart, R., and Daroczi, G. (2013), *Quandl: Quandl Data Connection*. R package version 2.1.2.

Ryan, J. A. (2015), *quantmod: Quantitative Financial Modelling Framework*. R package version 0.4-4.

Székely, G. J., and Rizzo, M. L. (2005), "Hierarchical Clustering Via Joint Between-Within Distances: Extending Ward's Minimum Variance Method," *Journal of Classification*, 22(2), 151 – 183.