

Teoria da Informação e Estatística Computacional no Processamento e Análise de Sinais

Implementações otimizadas em R e C

Aluna: Eduarda Tatiane Caetano Chagas

Orientador: Alejandro Cesar Frery Orgambide

**LaCCAN – Laboratório de análise e
computação científica**





1.

Séries temporels



Séries temporais

Tratam-se de conjuntos de dados, obtidos por meio de um processo observacional ao longo de um determinado período de tempo

Áreas de aplicação

- Bolsa de valores
- Medicina
- Meteorologia
- Cotação de commodities

Objetivo do projeto


Desenvolver um sistema portátil e interativo de análise de séries temporais





2.

Séries temporais e a Teoria da Informação



Etapas do processo de análise

Simbolização

◎ Processo de simbolização de Bandt e Pompe

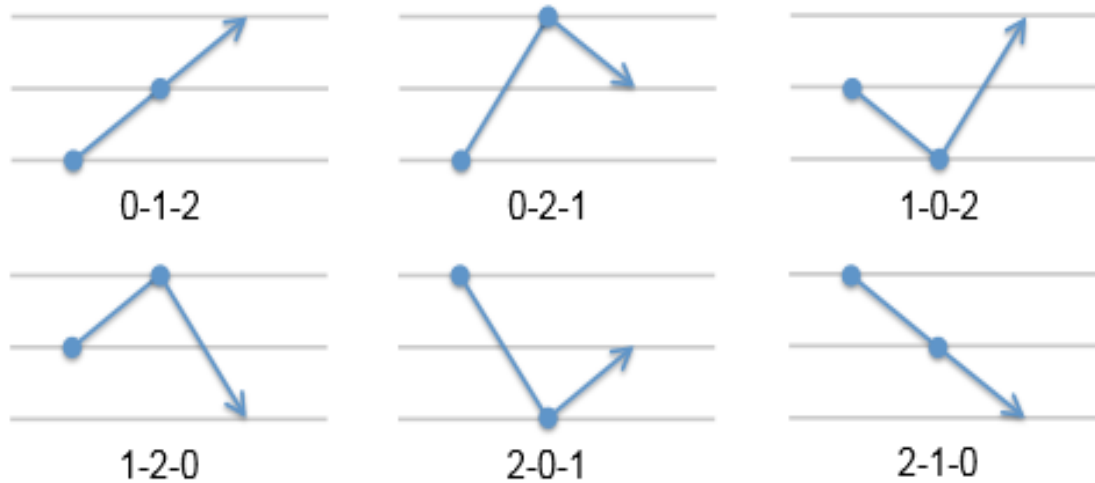
Extração de informações

◎ Entropias

◎ Distâncias estocásticas

◎ Complexidade estatística

Simbolização de Bandt e Pompe



Histograma

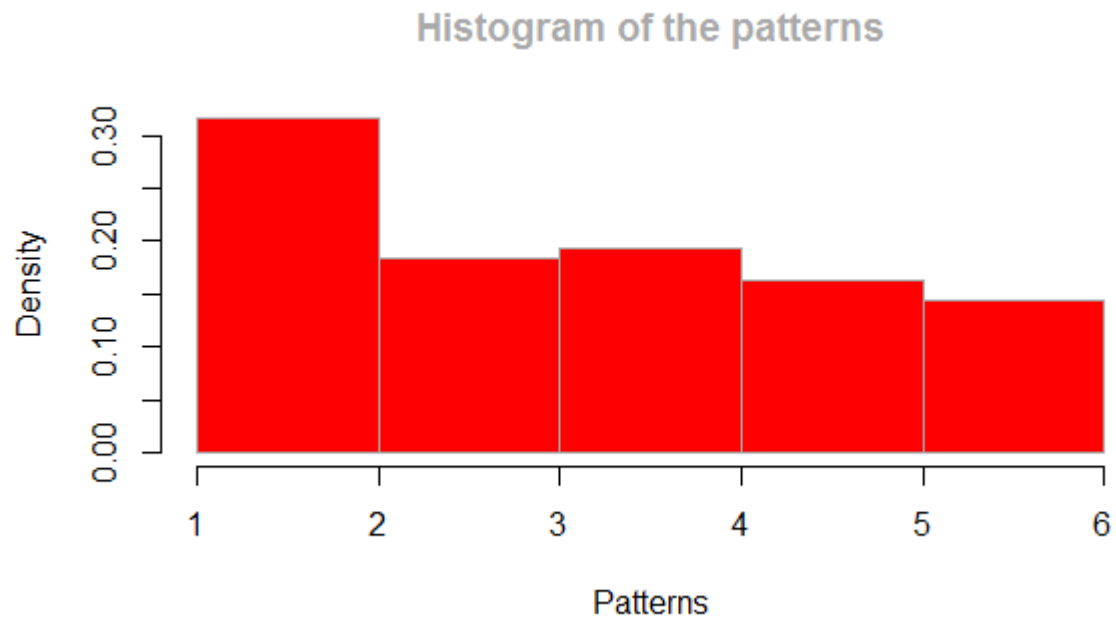
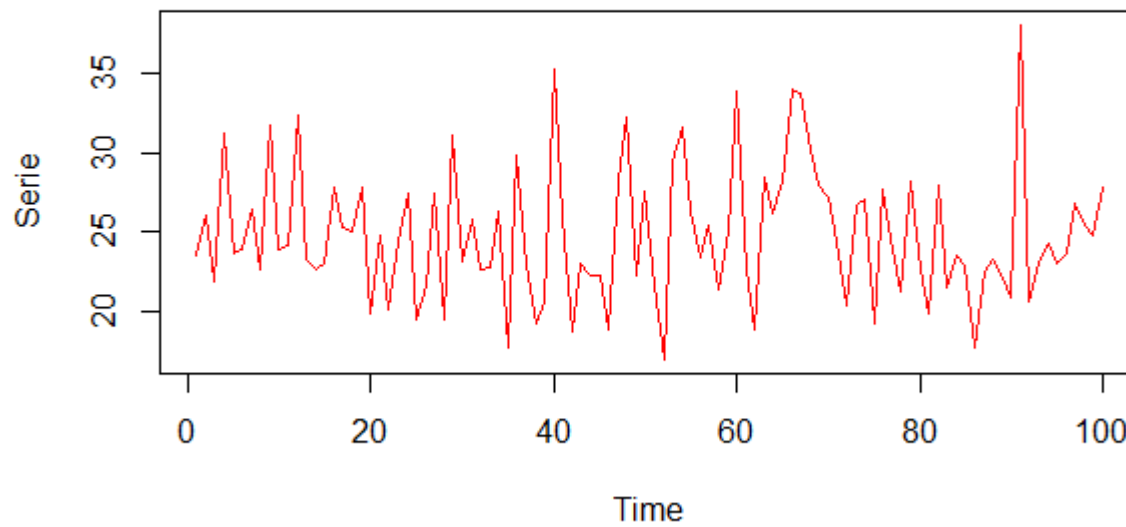


Gráfico da série

Graphic of time serie





“

Entropia

Corresponde à medida quantitativa de incerteza de uma estrutura descrita por uma distribuição de probabilidade

Entropia de Shannon

```
shannonEntropy <- function(probability){  
  h <- probability * log(probability)  
  h[is.nan(h)] <- 0  
  return (-sum(h))  
}
```

```
shannonEntropyNormalized <- function(probability){  
  return(H(probability)/log(length(probability)))  
}
```

Entropia de Tsallis

```
tsallisEntropy <- function(probability,q,option=0){  
  entropy = sum(probability^q)  
  entropy = (1 - entropy)^(1/(q - 1))  
  if(option){  
    return (entropy)  
  }  
  else{  
    ent_max = (1 - (length(probability)^(1 - q)))/(q - 1)  
    return( entropy/ent_max)  
  }  
}
```

Entropia de Renyi

```
renyiEntropy <- function(probability,q,option=0){  
  probability = distribution(initial,end)  
  entropy = sum(probability^q)  
  entropy = log(entropy)  
  entropy = entropy * (1/(1 - q))  
  if(option){  
    return (entropy)  
  }  
  else{  
    return ( entropy/log(length(probability)))  
  }  
}
```



“

Distância estocástica

Mensurando a similaridade entre duas séries temporais, tal medida é calculada através da análise de suas respectivas distribuições de probabilidades

Distâncias estocásticas

```
euclidian_distance<-function(probability){  
  c = rep(1/length(probability),length(probability))  
  distance = sum((probability-c)^2)  
  return(sqrt(distance))  
}
```

```
euclidian_quadratica_distance<-function(probability){  
  c = rep(1/length(probability),length(probability))  
  distance = sum((probability-c)^2)  
  return(distance)  
}
```

```
manhattan_distance<-function(probability){  
  c = rep(1/length(probability),length(probability))  
  distance = sum(abs(probability-c))  
  return(distance)  
}
```

Distâncias estocásticas

```
chebyshev_distance<-function(probability){  
  c = rep(1/length(probability),length(probability))  
  L = abs(probability - c)  
  return(max(L))  
}
```

```
kullback_leibler_divergence<-function(probability){  
  c = rep(1/length(probability),length(probability))  
  distance <- probability * log(probability/c)  
  distance[is.nan(distance)] <- 0  
  return(sum(distance))  
}
```

```
kullbach_aux<-function(p,q){  
  distance <- p * log(p/q)  
  distance[is.nan(distance) || is.infinite(distance)] <- 0  
  return(sum(distance))  
}
```


Distâncias estocásticas

```
hellinger_Distance<-function(probability){  
  c = rep(1/length(probability),length(probability))  
  distance = sum((sqrt(probability)-sqrt(c))^2)*0.5  
  return(sqrt(distance))  
}
```

```
jensenDivergence<-function(p){  
  q = rep(1/length(p),length(p))  
  s_p = shannonEntropy(p)  
  s_q = shannonEntropy(q)  
  s_pq = shannonEntropy((p+q)/2)  
  divergence = sum( s_pq - (s_p/2) - (s_q/2))  
  return(divergence)  
}
```

Distâncias estocásticas

```
wootters_distance<-function(probability,q){  
  c = rep(1/length(probability),length(probability))  
  dis = sum(sqrt(probability*c))  
  dis = acos(dis)  
  return(dis)  
}
```

```
complexityF<-function(probability){  
  entropy = ShannonAux(probability)  
  qInitial = rep(1/length(probability),length(probability))  
  qInitial[1] = 1  
  disequilibrium = jensenDivergence(qInitial) * jensenDivergence(probability)  
  comp = entropy * disequilibrium  
  return(comp)  
}
```

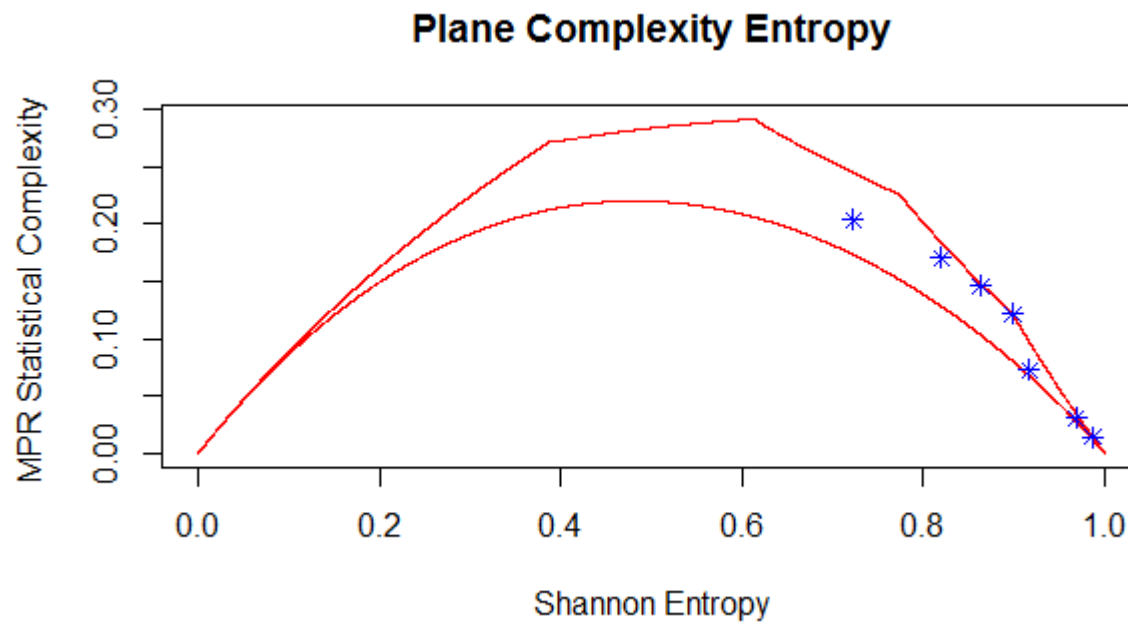


“

Complexidade estatística

*Procura encontrar estruturas de interação
de dependência entre os elementos de uma
dada série*

Plano Complexidade-Entropia



Planos futuros

Desenvolvimento de
tais metodologias no
ambiente
bidimensional,
realizando análises de
texturas (texturas de
Brodatz)



Referências

- Characterization of vehicle behavior with Information Theory / Aquino, A. L. L. et al. (2015)
- A Mathematical Theory of Communication / Shannon, C. E. (1948)
 - Measures of statistical complexity: Why? / Feldman and Crutcheld (1998)
- Permutation entropy: A natural complexity measure for time series / Bandt and Pompe (2002)



Obrigado !

Dúvidas?

