

# 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

December 7, 2016

LaCCAN – Laboratório de análise e computação científica

## 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

## Etapas do processo de análise

### Simbolização

*-Processo de simbolização de Bandt e Pompe*

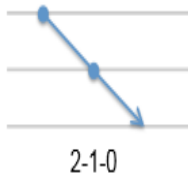
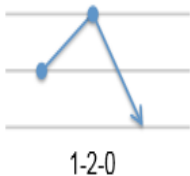
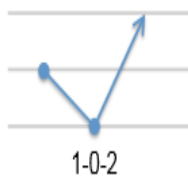
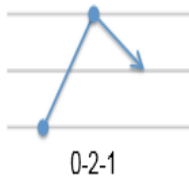
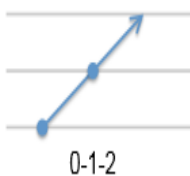
### Extração de informações

*-Entropias*

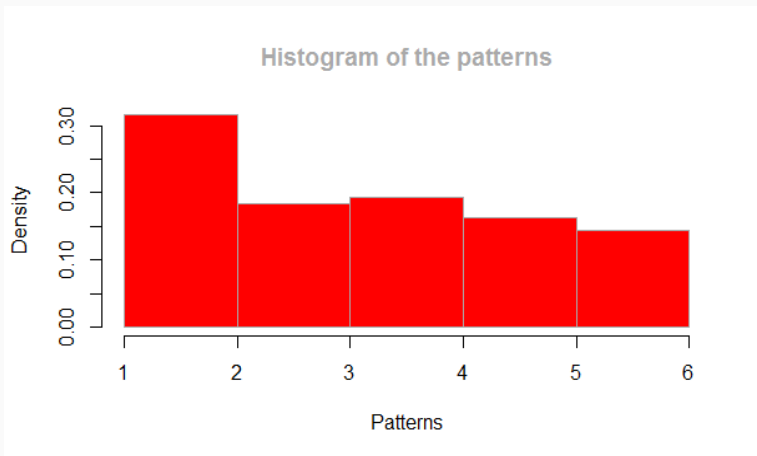
*-Distâncias estocásticas*

*-Complexidade estatística*

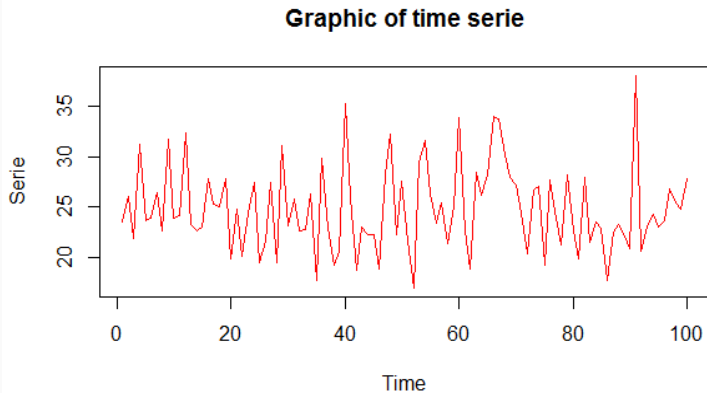
# SIMBOLIZAÇÃO DE BANDT E POMPE



# HISTOGRAMA



# GRÁFICO DA SÉRIE



## 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)] <-  
  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)  
}
```

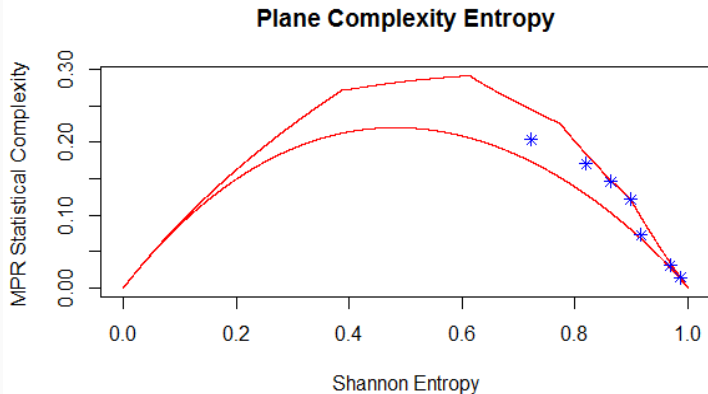
```
complexityF<-function(probability){  
  entropy = ShannonAux(probability)  
  qInitial = rep(1/length(probability),length(probability))  
  qInitial[1] = 1  
  disequilibrium = jensenDivergence(qInitial) * jensenDivergence(qInitial)  
  comp = entropy * disequilibrium  
  return(comp)  
}
```



## 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)

## References:

- 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 Crutchfield (1998)
- Permutation entropy: A natural complexity measure for time series / Bandt and Pompe (2002)

DÚVIDAS?