

# ANÁLISES EXPLORATÓRIAS DE DADOS

Neste tutorial, pretendemos instrumentalizar os(as) usuários(as) a realizar várias técnicas de Análise Exploratória de Dados (AED).

Apesar de existirem questionamentos estatísticos e filosóficos sobre a realização da AED antes das análises de dados previstas em um projeto de pesquisa, o contato prévio com os dados pode, no mínimo, auxiliar a detectar anomalias nos dados e buscar suas causas.

## Objetivos da Análise Exploratória de Dados (AED)

A AED teve origem a partir da expansão das análises frequentistas, porém, também pode ser utilizada no contexto de outras abordagens analíticas. As **premissas** que são avaliadas (quantitativamente ou graficamente) pela AED possuem ampla aplicação, são elas:

- A amostra foi obtida seguindo o princípio da aleatoriedade
- A amostra é proveniente de uma distribuição fixa (normal, binomial, etc)
- A distribuição tem uma “localização” fixa (média, lambda, etc)
- A distribuição tem variação fixa (variância, desvio, etc)

Dentre os principais **objetivos** de uma AED podemos listar os seguintes:

- Procurar variáveis mais importantes dentro de um conjunto abrangente;
- Compreender a estrutura implícita dos dados;
- Detectar pontos extremos (*outliers*) e anomalias;
- Testar premissas;
- Avaliar se os dados se ajustam aos modelos que serão utilizados nas análises;
- Determinar ajustes ótimos;

Alguns entusiastas da AED acreditam que muitas vezes é possível discutir os resultados obtidos a partir apenas da AED, sem precisar de testes de inferência estatística

Na página virtual [NIST - Engineering Statistics Handbook](#), são apresentadas várias questões que poderiam ser analisadas diretamente pela AED. Algumas são listadas abaixo:

- O que é um valor típico (p. ex., média, mediana, etc)?
- Qual é a incerteza para um valor típico (variância, desvio, etc)?
- Qual seria um bom ajuste (em relação às distribuições) para um dado conjunto de números?
- Quais são os valores de determinados percentis?
- Um determinado fator tem algum efeito?
- As medidas provenientes de diferentes fontes são equivalentes?
- Qual pode ser a melhor função para relacionar uma variável resposta a um conjunto de fatores?
- Podemos separar *signal* de *ruído* em dados temporalmente dependentes?
- Os dados têm valores extremos (*outliers*)?

Ao final deste tutorial esperamos que você consiga responder algumas dessas questões.

## Copiando os arquivos de dados e instalando pacotes

As análises serão realizadas em ambiente R e para isso teremos que instalar alguns pacotes, mas não será necessário ter conhecimento prévio sobre o R, pois forneceremos todos os comandos necessários para a realização da atividade.

1) Crie um diretório (pasta), copie os arquivos de dados abaixo para esse diretório e faça a descompactação no mesmo diretório:

- [autocorr.zip](#)
- univar.zip

2) Abra o R no seu computador e mude o diretório de trabalho para o diretório (*i.e.* a pasta) que você criou, usando o menu **Arquivo > mudar dir...**

3) Instale os pacotes *car* e *lattice*.

Para isso, basta copiar e colar os comandos que estão nas caixas de cor cinza:

```
install.packages("car")
```

Espere finalizar todo o processo de instalação desse pacote para iniciar o próximo:

```
install.packages("lattice")
```

4) Agora carregue os pacotes:

```
library(car)  
library(lattice)  
library(graphics)
```

## ANALISANDO DADOS UNIVARIADOS

### Conhecendo os dados:

1) importe o conjunto de dados para o R

```
univar1<- read.csv("univar1.csv")
```

2) Use a função *head* para visualizar as 5 primeiras linhas do conjunto de dados

```
head(univar1)
```

3) Inspecione o resumo dos dados

```
summary(univar1)
```

Note que para variáveis numéricas (contínuas ou discretas) são apresentados os valores Mínimo,

Máximo, Média, Mediana, Primeiro quartil, Terceiro quartil, e, no caso de haver dados faltantes, será apresentado o número de dados faltantes (linhas não preenchidas com valores), representados como "NA's". Para variáveis categóricas são apresentados os níveis existentes e quantas observações cada um dos níveis possui. Se houver dados faltantes, será apresentado o número de "NA's"

**Veja se você entendeu o conjunto de dados. Você consegue pensar em como esses dados podem estar distribuídos? Qualquer dúvida, pergunte!**

4) Se quiser, visualize o conjunto de dados como uma planilha convencional

```
edit(univar1)
```

## ANÁLISES GRÁFICAS



**Salve todos os gráficos que você criar a partir de agora**

1) Vamos olhar como os dados das quatro variáveis numéricas estão distribuídos graficamente. **1.a) Histograma de frequência:**

```
par(mfrow = c(2,2)) ##Aqui estamos criando um layout para colocar os quatro
gráficos juntos
hist(univar1$COMPRIMENTO_BICO)
hist(univar1$BIOMASSA_AVE)
hist(univar1$BIOMASSA_INSETOS)
hist(univar1$TAMANHO_SEMENTES)
par(mfrow=c(1,1)) ## voltando ao padrão de apresentar apenas 1 gráfico por
página
```

**O que está representado no eixo X e no eixo Y de cada um desses gráficos? Quais são os valores que definem as classes usadas no eixo X desse histograma? São os mesmos valores para todos os gráficos? Poderiam ser usados diferentes valores para essas mesmas variáveis? Qual a melhor forma de definir as classes de um histograma?**

E se mudássemos para 20 classes no eixo X, como ficariam os gráficos?

```
par(mfrow = c(2,2))
hist(univar1$COMPRIMENTO_BICO, breaks = 20)
hist(univar1$BIOMASSA_AVE, breaks = 20)
```

```
hist(univar1$BIOMASSA_INSETOS, breaks = 20)
hist(univar1$TAMANHO_SEMENTES, breaks = 20)
par(mfrow=c(1,1))
```

E se mudássemos para 10 classes no eixo X, como ficariam os gráficos?

```
par(mfrow = c(2,2))
hist(univar1$COMPRIMENTO_BICO, breaks = 10)
hist(univar1$BIOMASSA_AVE, breaks = 10)
hist(univar1$BIOMASSA_INSETOS, breaks = 10)
hist(univar1$TAMANHO_SEMENTES, breaks = 10)
par(mfrow=c(1,1))
```

**1.b) Gráfico de densidade** Ao invés de usarmos classes, podemos representar a distribuição por meio de uma linha, que é obtida usando a densidade estimada (por uma função conhecida como *kernel*) de valores para “janelas” (*bandwidth*) muito pequenas. Vamos ver como ficam as distribuições das nossas quatro variáveis numéricas:

```
par(mfrow = c(2,2))
plot(density(univar1$COMPRIMENTO_BICO))
plot(density(univar1$BIOMASSA_AVE))
plot(density(univar1$BIOMASSA_INSETOS))
plot(density(univar1$TAMANHO_SEMENTES))
par(mfrow=c(1,1))
```

Podemos juntar esses dois gráficos em um só. Para isso, use o código abaixo:

```
par(mfrow = c(2,2))
hist(univar1$COMPRIMENTO_BICO, prob=T )
lines(density(univar1$COMPRIMENTO_BICO))

hist(univar1$BIOMASSA_AVE, prob=T)
lines(density(univar1$BIOMASSA_AVE))

hist(univar1$BIOMASSA_INSETOS, prob=T)
lines(density(univar1$BIOMASSA_INSETOS))

hist(univar1$TAMANHO_SEMENTES, prob=T)
lines(density(univar1$TAMANHO_SEMENTES))
par(mfrow=c(1,1))
```

Podemos também mostrar, na parte inferior do gráfico de densidade, o número de observações em cada faixa do gráfico. Para isso vamos usar a função *rug()*

```
par(mfrow = c(2,2))
plot(density(univar1$COMPRIMENTO_BICO))
rug(univar1$COMPRIMENTO_BICO, side=1)

plot(density(univar1$BIOMASSA_AVE))
rug(univar1$BIOMASSA_AVE, side=1)
```

```
plot(density(univar1$BIOMASSA_INSETOS))
rug(univar1$BIOMASSA_INSETOS, side=1)

plot(density(univar1$TAMANHO_SEMENTES))
rug(univar1$TAMANHO_SEMENTES, side=1)

par(mfrow=c(1,1))
```

Todas essas informações nos auxiliam para identificarmos a quais distribuições teóricas nossos dados se ajustam.

**1.c) Box-plot ou Box-whiskers plot ou Five-numbers-summary** Um box-plot clássico utiliza os seguintes valores:

- - Mínimo
- - Primeiro quartil
- - Mediana
- - Terceiro quartil
- - Máximo

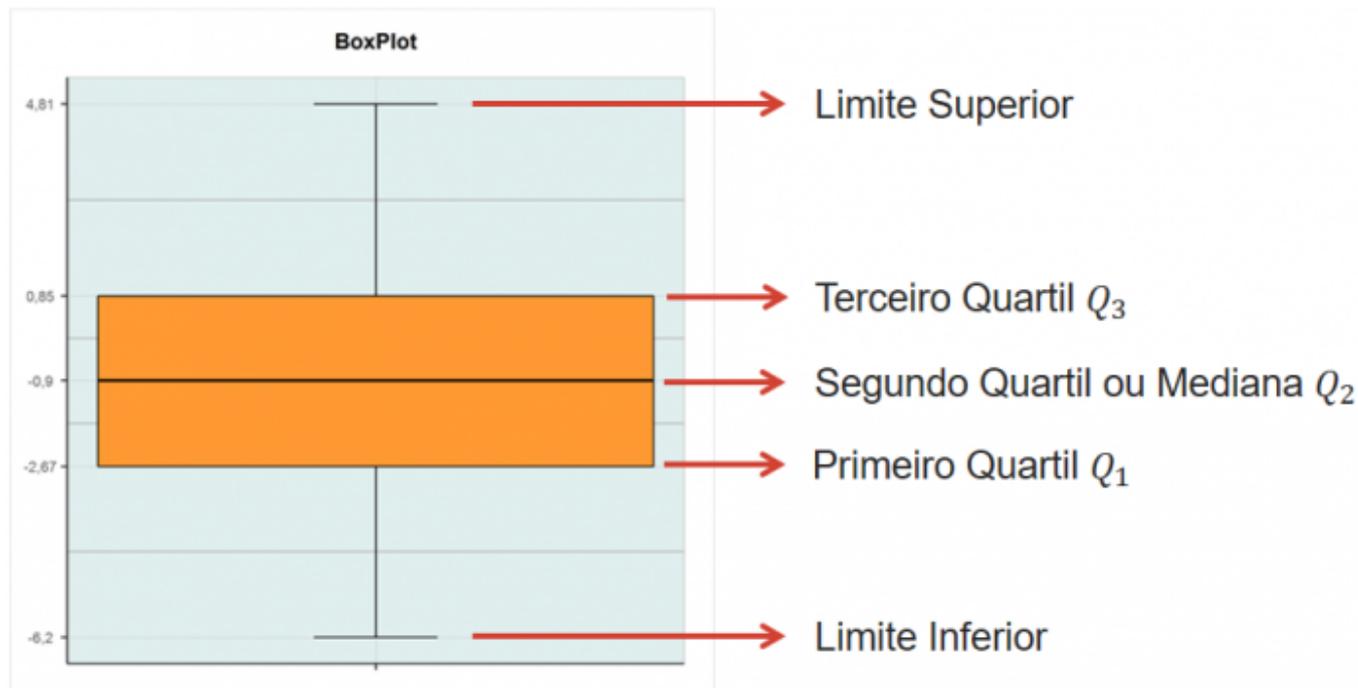
- Ordene a variável COMPRIMENTO\_BICO do menor para o maior valor:

```
(sort(univar1$COMPRIMENTO_BICO))
```

- - Anote quantos dados existem (i.e. qual é o  $n$  da amostra)?
- - Anote os valores mínimo e o máximo
- - Anote o valor que separa os dados ordenados em duas metades (i.e. o valor que representa 50% dos dados).
- - Anote o valor que separa os primeiros 25% valores dos restantes 75%
- - Anote o valor que separa os primeiros 75% valores dos restantes 25%

 Quando o quartil desejado (primeiro, segundo [= mediana] ou terceiro) se posicionar exatamente sobre um dado valor, use esse valor. Quando o quartil desejado se localizar entre dois valores, retire a média desses dois valores.

Com esses dados você poderia construir um box-plot manualmente, conforme a figura abaixo:



Mas temos uma função que faz isso por nós:

```
boxplot(univar1$COMPRIMENTO_BICO, range=0)
```

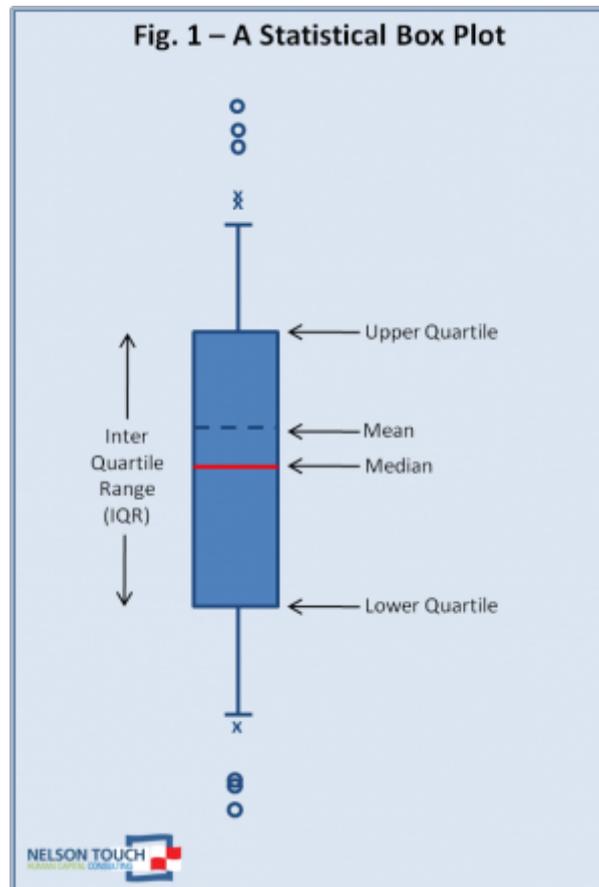
Confira se os valores utilizados pela função *boxplot* são iguais aos que você calculou

Muitas vezes, o *default* de um programa estatístico faz um gráfico chamado **boxplot modificado**. Esse **boxplot modificado** nos ajuda a identificar os pontos extremos que comumente chamamos de **outliers**.

Ao invés de usarmos os valores de **máximo** e **mínimo** nas pontas das linhas verticais (tanto para cima quanto para baixo), usamos a seguinte equação  $1.5 \cdot IQ$  para definirmos o comprimento da linha vertical.  $IQ$  é a distância entre o primeiro e o terceiro quartil (ou distância interquartil ou ainda a amplitude da caixa central do boxplot).  $IQ = Q_3 - Q_1$ , onde  $Q_3$  é o valor do terceiro quartil e  $Q_1$  é o valor do primeiro quartil.

Suponha uma situação em que sua variável resposta é medida em ml. Se  $Q_3=30\text{ml}$  e  $Q_1=20\text{ml}$ , então,  $IQ = 10\text{ml}$  e a linha vertical deve ter  $15\text{ml}$  ( $IQ = 1.5 \cdot 10$ ). Como a linha vertical é plotada a partir das bordas das caixas (quartis) para se obter o valor superior da linha vertical é preciso somar  $Q_3 + IQ$  ( $30 + 15 = 45\text{ml}$ ) e para se obter o valor inferior da linha vertical é preciso subtrair  $Q_1 - IQ$  ( $20 - 15 = 5\text{ml}$ ). Os valores que estiverem abaixo de  $5\text{ml}$  e acima de  $45\text{ml}$  serão considerados *outliers*.

Um boxplot modificado fica assim:



O valor a ser multiplicado por IQ pode variar de um autor para outro e de um programa computacional para outro, então é muito importante que as legendas dos gráficos tragam essa informação. Infelizmente, essa não é uma prática comum.

Vamos fazer um boxplot modificado com os nossos dados de COMPRIMENTO\_BICO

```
boxplot(univar1$COMPRIMENTO_BICO)
```

**Várias informações podem ser obtidas a partir de um boxplot. - Existem *outliers* no conjunto de dados? Eles estão entre os valores mais altos ou mais baixos? - A distribuição dos dados é simétrica ou assimétrica? - Se for assimétrica, os dados estão concentrados em valores mais altos (assimetria com cauda grande para a esquerda) ou mais baixos (assimetria com cauda grande para a direita)? \ \* PARA PENSAR: É possível construir um box-plot com menos de 5 dados?**

### 1.d) Comparando Boxplots (e usando o argumento *notched*)

Podemos comparar visualmente as respostas de organismos a dois níveis de um determinado tratamento, ou de uma determinada condição (variáveis categóricas), analisando os boxplots de cada conjunto de dados.

No nosso conjunto de dados, podemos avaliar se a biomassa de insetos apresenta diferentes distribuições em locais com diferentes níveis de distúrbio:

```
boxplot(univar1$BIOMASSA_INSETOS ~ univar1$NIVEL_DISTURBIO)
```

**Avalie a posição das medianas, a distribuição dos valores entre os quartis, os valores máximos e mínimos, a simetria da distribuição, os outliers e indique se, a partir desses gráficos, você se sente confiante em afirmar se a biomassa de insetos difere ou não entre os dois níveis de distúrbio.**

Existe uma forma bastante simples de calcular “**intervalos de confiança da mediana**” e que podem nos ajudar a tomar decisões a respeito da similaridade ou diferença de respostas. Esse método é baseado em técnicas de Monte Carlo<sup>1)</sup> e produz os valores limites para os intervalos de confiança da mediana. Existe um argumento (*notched*) que podemos inserir na função *boxplot()*, que permite que esses intervalos de confiança da mediana sejam visualizados como um estreitamento da caixa central do boxplot na região que estiver dentro do intervalo de confiança da mediana. Se esses estreitamentos não se sobrepuserem podemos admitir que os dois conjuntos de dados são diferentes.

```
boxplot(univar1$BIOMASSA_INSETOS ~ univar1$NIVEL_DISTURBIO, notch=TRUE)
```

**E agora, você está mais seguro(a) para afirmar se a biomassa de insetos difere ou não entre os dois níveis de distúrbio?**

## CHECANDO O AJUSTE DOS DADOS A UMA DISTRIBUIÇÃO

Agora vamos avaliar visualmente se as variáveis se distribuem de acordo com uma distribuição conhecida. Por simplicidade, vamos fazer isso com a **distribuição normal**, porém, o mesmo pode ser feito com outros tipos de distribuição.

IMPORTANTE: Vários tipos de análises têm a normalidade como premissa. Porém, é importante não confundir a normalidade dos dados brutos, com a normalidade dos erros, da variância ou dos resíduos das relações. No tópico **Testes clássicos frequentistas** vamos entender o que significa analisar a normalidade dos resíduos. Nesse momento, em que estamos analisando a normalidade de uma variável isoladamente, estamos apenas compreendendo e descrevendo o tipo de dado que coletamos.

## Gráfico quantil-quantil

A ideia de um gráfico quantil-quantil é expressar visualmente o quanto seus dados se aproximam de

uma determinada distribuição. Eles podem ser usados para comparar as distribuições de dois conjuntos de dados diferentes (para saber se ambos vêm de uma mesma população) ou para comparar a distribuição de um conjunto de dados coletados com uma distribuição de probabilidade teórica (normal, binomial, etc). Nesse segundo caso, eles são também chamados de Gráficos de Probabilidade (*Probability Plots*). No exemplo abaixo, vamos comparar dados coletados e uma distribuição normal.

Um quantil<sup>2)</sup> representa a porcentagem de dados (ordenados) que estão abaixo de um dado valor. Por exemplo, um quantil de 0.4 (ou 40%) é o ponto no qual 40% dos dados ocorrem abaixo do valor correspondente e 60% dos dados ocorrem acima desse valor.

**Relembrando: em uma distribuição normal, qual o valor que limita os dados abaixo de 2.5%? Expresse em termos de desvio-padrão para facilitar**

Para facilitar a comparação do valor que representa, por exemplo, 10% de uma distribuição normal e comparar com o valor que representa 10% dos seus dados, tanto a distribuição normal quanto seus dados precisam estar na mesma escala. Então, para que os dois eixos do gráfico estejam exatamente na mesma escala, os quantis esperados para uma distribuição normal são distribuídos ao longo da amplitude (mínimo e máximo) dos dados coletados.

No eixo X serão projetados os valores esperados pela distribuição normal para cada quantil e no eixo Y serão projetados os valores dos dados coletados, também para cada quantil. A maior parte dos programas traça uma linha diagonal em 45 graus, para auxiliar a visualização.

Vamos então aplicar as funções abaixo aos nossos dados:

```
par(mfrow = c(2,2))

qqnorm(univar1$COMPRIMENTO_BICO)
qqline(univar1$COMPRIMENTO_BICO)

qqnorm(univar1$BIOMASSA_AVE)
qqline(univar1$BIOMASSA_AVE)

qqnorm(univar1$BIOMASSA_INSETOS)
qqline(univar1$BIOMASSA_INSETOS)

qqnorm(univar1$TAMANHO_SEMENTES)
qqline(univar1$TAMANHO_SEMENTES)

par(mfrow=c(1,1))
```

**Uma vez compreendida a forma como esse gráfico foi construído, como os resultados devem ser interpretados? Para cada uma das variáveis avalie: - É possível visualizar *outliers* nas variáveis? - A distribuição dos dados é simétrica ou assimétrica? Se for assimétrica, os dados estão**

**concentrados em valores mais altos (assimetria com cauda grande para a esquerda) ou mais baixos (assimetria com cauda grande para a direita)? - Os dados se ajustam bem à distribuição normal?**

## AVALIANDO AUTOCORRELAÇÃO

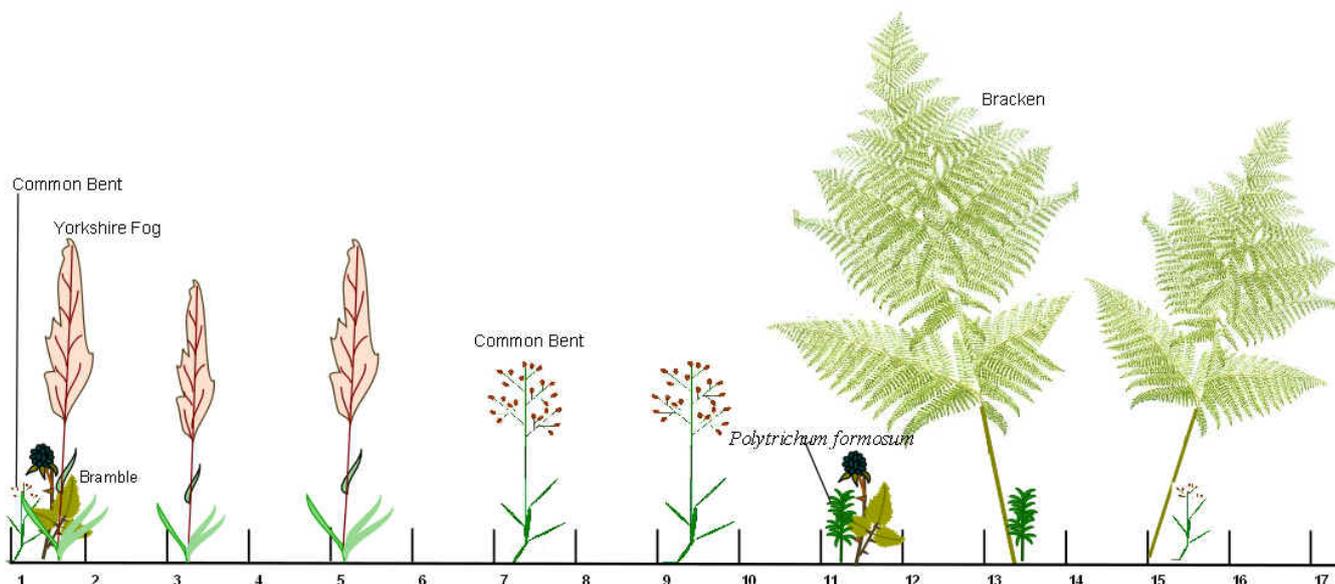
Para essa parte do tutorial, importe o conjunto de dados "autocorr.csv" para o R e inspecione os dados:

```
autocorr<- read.csv("autocorr.csv")  
head(autocorr)  
summary(autocorr)
```

Dentre as premissas mais importantes dos testes estatísticos está a **independência** (espacial e/ou temporal) dos dados coletados. Existem diversas formas de avaliar o nível de autocorrelação<sup>3)</sup> entre os dados. Quando estamos lidando com dados distribuídos em apenas uma dimensão (transectos lineares ou series temporais direcionais), esse processo é mais simples. Porém, quando os dados estão distribuídos em duas (p.ex. posições x e y em uma parcela) ou em três (posições x e y e mais a profundidade, no caso de medidas em sistemas aquáticos) dimensões os métodos são mais complexos e fogem ao escopo desse tutorial. Se tiver interesse em entender alguns desses métodos para duas dimensões visite [Padrões Multiescala](#)

Porém, existe uma forma simples de visualizar os dados e obter uma primeira impressão sobre possíveis autocorrelações para dados coletados em transectos lineares e também para dados de séries temporais.

Imagine o transecto abaixo:



Você poderia se perguntar se os dados mais próximos espacialmente são mais parecidos entre si (i.e. positivamente autocorrelacionados). Uma forma de avaliar isso é plotar o valor de um dado em relação ao seu antecessor, então, no eixo X teríamos os valores do segundo dado em diante e no eixo Y teríamos, correspondendo a cada valor do eixo X, o valor do dado anterior. Esse tipo de gráfico é chamado de **lag plot**

O gráfico do tipo *lag-plot* apresenta a relação entre um determinado dado e o seu antecessor (temporal ou espacial) **quando os dados são tomados em uma sequência unidimensional**.

O que você esperaria que acontecesse em um gráfico desse tipo, caso os valores estejam autocorrelacionados? E se não estiverem?

Vejam como ficam esses gráficos para os dados do conjunto **autocorr** que temos disponível para essa análise. Nesse arquivo temos os dados de dois transectos (x1 e x2) com 100 pontos cada.

```
lag.plot(autocorr$x1, do.lines = FALSE, diag=FALSE)
```

```
lag.plot(autocorr$x2, do.lines = FALSE, diag=FALSE)
```

**Olhando para esses resultados, qual a sua conclusão?**

Entretanto, é importante entender que no gráfico padrão (*default*) produzido por essa função *lag.plot()* estamos relacionando um determinado dado com o seu antecessor **imediate**, ou seja, o antecessor está a 1 unidade de distância em relação ao dado que colocamos no eixo X. Porém, alguns processos podem ocorrer em escalas diferentes de 1 unidade de distância e podemos querer checar se existe autocorrelação em outras escalas. Para isso, usamos o argumentos "*lags*" e "*set.lags*" dentro da função *lag.plot()*.

Vamos ver como ficam os gráficos com `lags=2`

```
lag.plot(autocorr$x1, do.lines = FALSE, lags=2, set.lags=2, diag=FALSE)
```

```
lag.plot(autocorr$x2, do.lines = FALSE, lags=2, set.lags=2, diag=FALSE)
```

**E então, as conclusões se mantêm?**

## ANALISANDO DADOS BIVARIADOS

Muitas vezes não estamos interessados em analisar a distribuição de uma variável *per se*, mas sim em analisar se existe alguma relação entre duas variáveis numéricas ordenadas.

Alguns pontos que queremos analisar quando testamos uma relação entre variáveis são:

- 1 - Qual a direção da relação (positiva ou negativa)?
- 2 - A relação é linear (i.e. para cada aumento na variável X, a variável Y aumenta a uma taxa constante)?
- 3 - Como os valores da variável do eixo Y variam em relação aos valores da variável do eixo X (muita ou pouca variação nos valores de Y para valores similares de X)?
- 4 - A variação de Y é similar ao longo de todo o eixo X?

Para analisarmos isso visualmente, vamos construir um **gráfico de dispersão**.

Usaremos um conjunto de dados diferente para essas análises. - Copie e descompacte o arquivo abaixo no seu diretório

- [bivar.zip](#)

-Importe o arquivo para o R

```
bivar<-read.csv("bivar.csv")  
head(bivar)  
summary(bivar)
```

Faça um gráfico de dispersão (ou *Gráfico XY*) e descreva sua primeira impressão sobre a relação entre as variáveis.

```
plot(bivar$y.l ~ bivar$x.l)
```

**Analisando esses gráficos, você conseguiria responder às 4 questões colocadas no início dessa seção para cada um deles?**

Para tentar captar a tendência da relação, você poderia ir traçando pequenas linhas que buscassem a melhor relação entre os dados da variável y.l e da variável x.l ao longo de pequenos trechos da variável x.l, como se estivesse desenhando “à mão”. Existe uma função que faz isso. Ela se chama *lowess* e pode ser aplicada da seguinte forma:

```
plot(bivar$y.l ~ bivar$x.l)  
lines(lowess(bivar$y.l ~ bivar$x.l))
```

**E agora, suas respostas às 4 questões colocadas no início dessa seção mudariam?**

Agora vamos fazer o gráfico de dispersão para outras duas variáveis y.n (resposta) e x.n (preditora):

```
plot(bivar$y.n ~ bivar$x.n)  
lines(lowess(bivar$y.n ~ bivar$x.n))
```

**Analisando esse gráfico, como você responderia às 4 questões colocadas no início dessa seção?**

Existe um gráfico no pacote *car* que nos mostra várias informações de uma relação entre duas variáveis e pode ajudar bastante no entendimento da relação.

```
scatterplot (bivar$y.l ~ bivar$x.l)
```

```
scatterplot (bivar$y.n ~ bivar$x.n)
```

## Transformando os dados:

Algumas vezes, a relação que observamos entre duas variáveis não é linear, mas gostaríamos de analisar essa relação dentro do escopo de uma Análise de Regressão Linear, em função das facilidades de trabalhar com esse tipo de análise. Para isso, precisamos recorrer aos recursos de **transformação dos dados**.

Esses recursos podem ser utilizados para fazer com que a distribuição dos dados de uma variável (ou de ambas) seja mais similar a uma distribuição normal.

ATENÇÃO: Atualmente, existem muitas formas alternativas de realizar as análises sem que haja necessidade de transformação dos dados (ver o tópico *Modelos Lineares Generalizados*). Porém, para esse tutorial, vamos analisar o que acontece quando usamos algumas transformações básicas.

### Logaritmo natural (ln)

Vamos analisar a relação entre as variáveis `COMPRIMENTO_BICO` e `BIOMASSA_AVE` e verificar se a relação parece linear. Para isso vamos utilizar o gráfico síntese produzido pelo pacote *car*:

```
scatterplot(univar1$COMPRIMENTO_BICO ~ univar1$BIOMASSA_AVE)
```

Como podemos observar pelos boxplots laterais, nesse caso, aparentemente são os dados da variável Y que parecem estar afetando a linearidade da relação. Então, vamos transformar os dados de Y pelo logaritmo natural e ver se o ajuste melhora.

```
scatterplot (log(univar1$COMPRIMENTO_BICO) ~ univar1$BIOMASSA_AVE)
```

**E agora, a relação parece mais linear? E os dados de `log(univar1$COMPRIMENTO_BICO)`, estão distribuídos de forma mais similar a uma distribuição normal?**

**Cuidado! Agora a relação linear é entre “log(Comprimento de bico)” e a “Biomassa das aves”, então é o “Logaritmo do Comprimento de bico” que aumenta a uma taxa constante em relação à “Biomassa das aves” e não mais o “Comprimento de bico”.**

Outras transformações que podem ser utilizadas: ) logaritmo base 10 (também para variáveis contínuas) ) logaritmo natural de  $x+1$  (quando a variável tem muitos zeros) ) raiz quadrada (para variáveis que representam contagens, p.ex.: número de indivíduos) ) arco seno (para variáveis que representam proporções/porcentagens)

1)

ver tópico de randomização e permutação

2)

Os **quartis** calculados anteriormente, são casos especiais de quantis. São chamados de quartis, pois dividem os dados em 4 grupos com a mesma proporção.

3)

*“Note that uncorrelated does not necessarily mean random. Data that has significant autocorrelation is not random. However, data that does not show significant autocorrelation can still exhibit non-randomness in other ways. Autocorrelation is just one measure of randomness.”*

From:

<http://labtrop.ib.usp.br/> - **Laboratório de Ecologia de Florestas Tropicais**

Permanent link:

<http://labtrop.ib.usp.br/doku.php?id=cursos:planeco2017:roteiro:05-descr>



Last update: **2018/03/05 12:12**