ESTATÍSTICA E ANÁLISE DE DADOS ESPACIAIS NO R: UM ESTUDO DE CASO COM DADOS DO LAGO WALKER

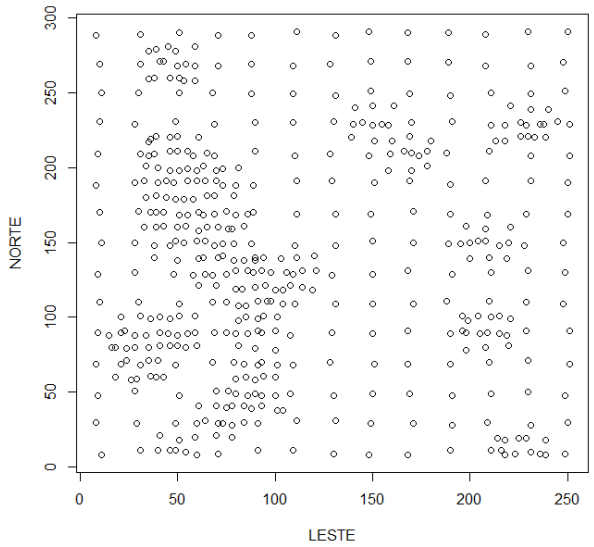
Jorge Kazuo Yamamoto

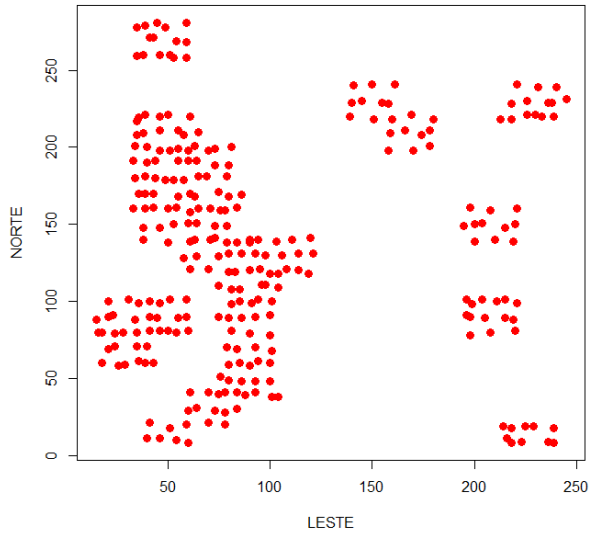
Professor Titular aposentado do Instituto de Geociências – USP. Atualmente, Professor Sênior do Departamento de Engenharia de Minas e de Petróleo – Escola Politécnica – USP.

GEO KRIGAGEM

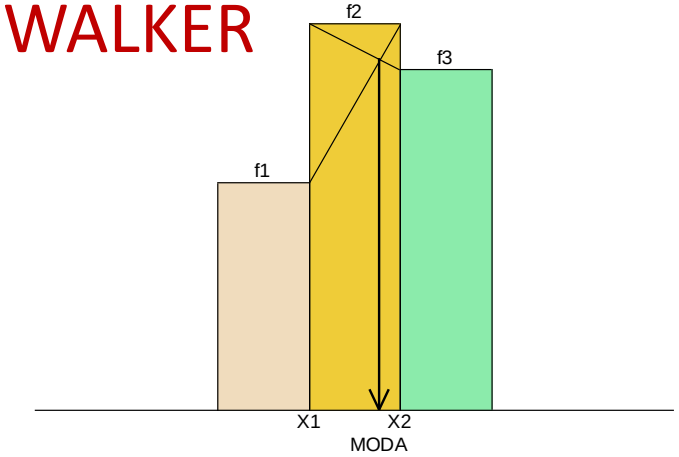
GEOCAST BRASIL

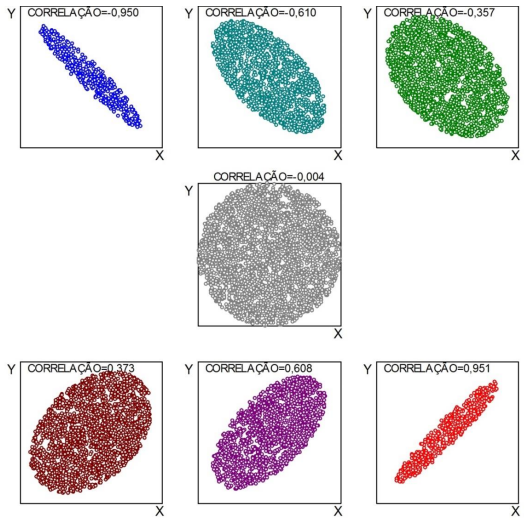
ESTATÍSTICA E ANÁLISE DE DADOS ESPACIAIS NO R:
UM ESTUDO DE CASO COM DADOS DO LAGO WALKER

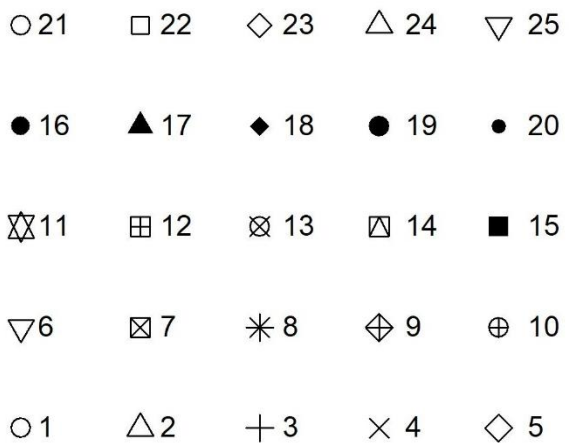


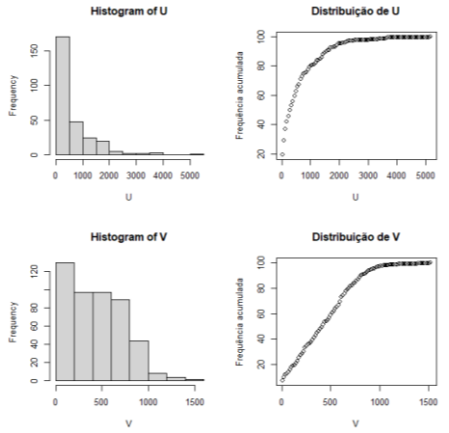


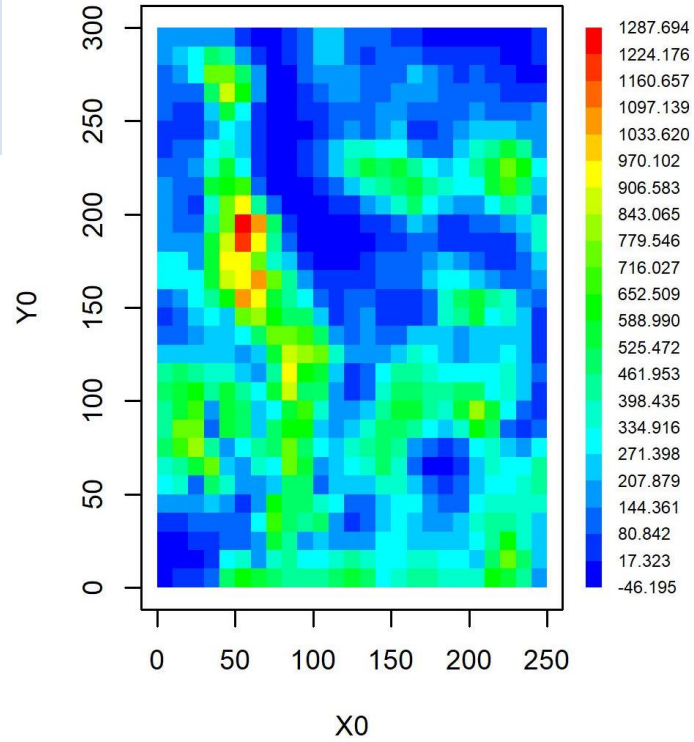
	estats	valores
	"No. de dados"	"470"
	"Média"	"435.299"
	"Mediana"	"424"
	"Moda"	"63.671"
	"Variância"	"89929.395"
	"Desvio Padrão"	"299.882"
	"Coef. de variação"	"0.689"
	"Amplitude interquartil"	"456.25"
	"Assimetria"	"0.459"
	"Curtose"	"2.871"
	"Valor mínimo"	"0"
25%	"Quartil Inferior"	"184.6"
75%	"Quartil Superior"	"640.85"
	"Valor máximo"	"1528.1"











Por que aprender uma linguagem de programação?

- Tornar-se independente de programas comerciais fechados;
- Ajudar na solução de problemas e desenvolver habilidade lógica;
- Permitir a coleta, gerenciamento e análise de dados;
- Transformar dados de saída de um programa para a entrada de outro;
- Manipular estruturas de dados;
- Fazer cálculos estatísticos e matemáticos;
- Gerar gráficos de alta qualidade com forte impacto visual;
- Fazer a conexão entre a teoria e a prática;
- Trabalhar em análise de dados;
- Ampliar horizontes de atuação profissional;
- Tornar-se cientista de dados – setor multiprofissional – profissão do futuro;
- Adquirir formação complementar em computação, qualquer que seja a graduação (administração, economia, engenharia, direito, geologia, geotecnologias, biologia, ecologia, agronomia etc.)
- Escrever seus próprios programas é muito diferente de aprender a usar um determinado programa comercial ou qualquer programa open source.

GEO KRIGAGEM

Por onde começar...

- Escolher uma linguagem de preferência open source;
- Usar programa open source significa estar completamente livre do pagamento de licenças e taxas de manutenção;
- Valer-se de um sistema colaborativo com milhares de pacotes e funções disponíveis;
- Ter o suporte de uma comunidade aberta (https://stackoverflow.com/* e https://stackexchange.com/) onde poderá encontrar respostas para suas questões;
- Existem dezenas de linguagens de programação em plataformas open source, dentre as quais Python e R;
- Ambas as linguagens são excelentes, mas o Autor começou por aprender R há 3 anos e por experiência própria pode afirmar que tem uma curva de aprendizado rápido.

* Número de questões em 07/09/2019: 18.179.078 para todas as linguagens de programação!

GEO KRIGAGEM

O que é R*?

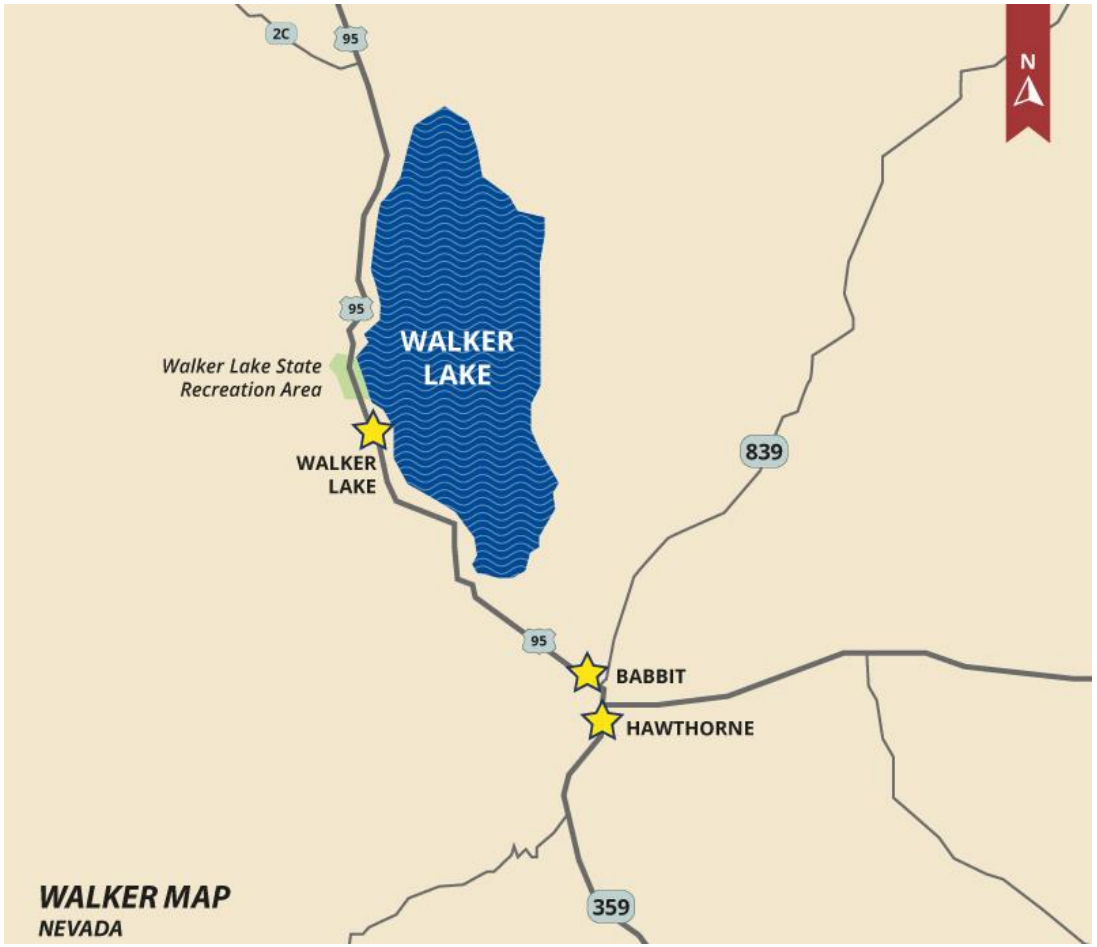
R é uma linguagem de programação e ambiente para computação estatística e gráfica. Trata-se um pacote integrado com facilidades para manipulação de dados, cálculos e representações gráficas. A linguagem R proporciona:

- Sistema efetivo para tratamento de dados e armazenamento;
- Conjunto de operadores para cálculos em arranjos, particularmente matrizes (programação orientada a objetos);
- Uma grande e coerente coleção de ferramentas intermediárias para análise de dados;
- Facilidades gráficas para análise de dados e representação no monitor ou em formatos gráficos (raster e vetorial);
- Uma linguagem de programação bem desenvolvida, simples e efetiva que inclui comandos condicionais, comandos de laço, funções definidas pelo usuário e facilidades de entrada e saída de dados.

* https://www.r-project.org/about.html

GEO KRIGAGEM

LOCALIZAÇÃO DO LAGO WALKER EM NEVADA (EUA)

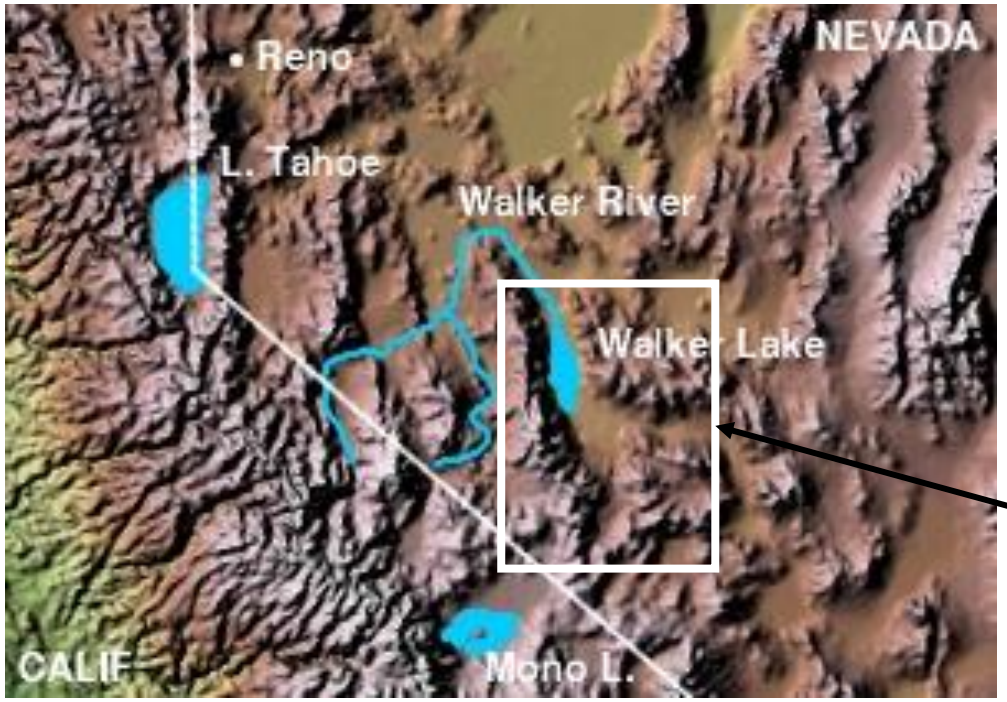


Fonte: https://www.easternsierrafishreports.com/lakes/59/walker-lake.php

GEOCAST
BRASIL

OBTENÇÃO DOS DADOS WALKER LAKE A PARTIR DA TOPOGRAFIA

Modelo de elevação de terreno



transformação*

Dados simulados V e U (ppm)
e T = variável indicadora*

Localização aproximada do walker.dat

Fonte: https://en.wikipedia.org/wiki/Walker_Lake_(Nevada)

*Segundo Isaaks e Srivastava (1989, p. 4-6).

Walker Lake Data

Isaaks e Srivastava (1989, p. 4-9) proporcionam o conjunto de dados walker.dat que representa uma amostra com 470 pontos, que foram extraídos de um conjunto completo (população) com 78000 pontos em uma malha regular de 260 por 300 nós. Esses dados estão salvos no arquivo walker_dat.csv, que contém seis variáveis: ID, Xlocation, Ylocation, V, U e T. V e U representam teores simulados em ppm (derivados da topografia do terreno) e T uma variável indicadora.

walker_dat-csv

```
1   ID;Xlocation;Ylocation;V;U;T
2   1;11;8;0.;-99.0;2
3   2;8;30;0.;-99.0;2
4   3;9;48;224.4;-99.0;2
5   4;8;69;434.4;-99.0;2
6   5;9;90;412.1;-99.0;2
7   6;10;110;587.2;-99.0;2
8   7;9;129;192.3;-99.0;2
9   8;11;150;31.3;-99.0;2
10  9;10;170;388.5;-99.0;2
11  10;8;188;174.6;-99.0;2
12  11;9;209;187.8;-99.0;2
13  12;10;231;82.1;-99.0;1
14  13;11;250;81.1;-99.0;1
15  14;10;269;124.3;-99.0;2
```

-99 indica dados não disponíveis para a variável U.

OBTENDO O MAPA DE LOCALIZAÇÃO DOS PONTOS DE DADOS

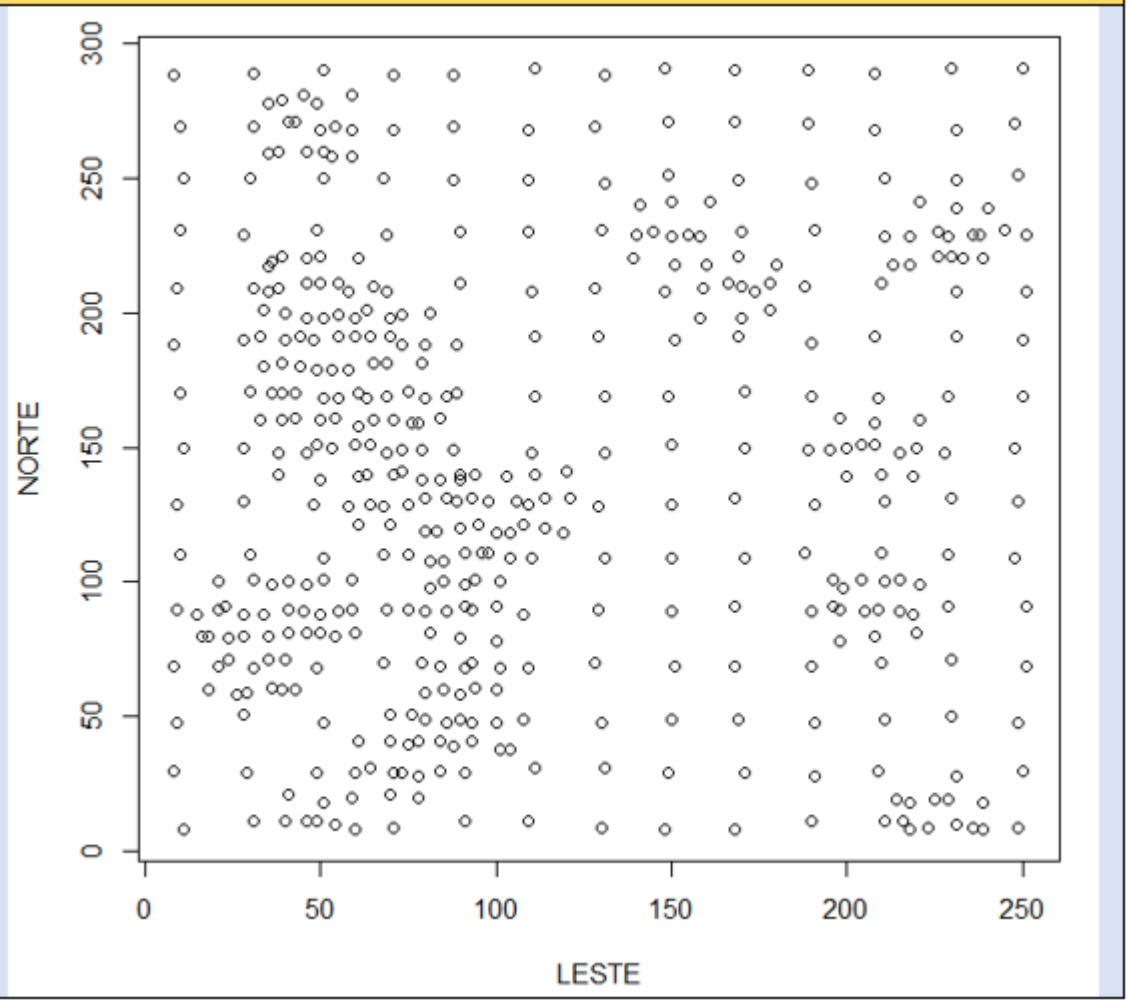
Script mapa_localizacao.R

```
1   > def.par=par(no.readonly=TRUE)
2   > setwd("C:\\IPT2021\\dados\\walker_data")
3   > dados=read.csv("walker_dat.csv",sep=";",header=T)
4   > x=dados$Xlocation; y=dados$Ylocation
5   > U=dados$U[dados$U!=-99.00];
6   > V=dados$V[dados$V!=-99.00]
7   > print(c(length(U),length(V),length(x)))
8   [1] 275 470 470
9   > plot(x,y,xlab="LESTE",ylab="NORTE")
10  > par(def.par)
```

Comentários:

- Os objetos U e V recebem todos os valores diferentes de -99 (linhas 5-6);
- Linha 7 usa a função print() com o comando de concatenação c();
- Na linha 8, temos que os objetos U e V têm tamanhos diferentes;
- Uso da função plot() para gerar o mapa (linha 9).

Dispositivo gráfico



Todos os scripts estão disponíveis em walker_lake_scripts.R – que serão encaminhados por e-mail para os interessados.

OBTENDO O MAPA DE LOCALIZAÇÃO DOS PONTOS DE DADOS

Como visto anteriormente, os objetos U e V têm tamanhos diferentes. Como faço o mapa somente com os pontos da variável U?

	A	B	C	D	E	F
1	ID	Xlocation	Ylocation	V	U	T
2	1	11	8	0	-99	2
3	2	8	30	0	-99	2
4	3	9	48	224.4	-99	2
5	4	8	69	434.4	-99	2
6	5	9	90	412.1	-99	2
7	6	10	110	587.2	-99	2
8	7	9	129	192.3	-99	2
9	8	11	150	31.3	-99	2
10	9	10	170	388.5	-99	2
11	10	8	188	174.6	-99	2
12	11	9	209	187.8	-99	2
13	12	10	231	82.1	-99	1
14	13	11	250	81.1	-99	1
15	14	10	269	124.3	-99	2
16	15	8	288	188	-99	2
17	16	31	11	28.7	-99	2
18	17	29	29	78.1	-99	2
19	18	28	51	292.1	-99	2
20	19	31	68	895.2	-99	2

-99 indica dados não disponíveis para a variável U.

OBTENDO O MAPA DE LOCALIZAÇÃO DOS PONTOS DE DADOS

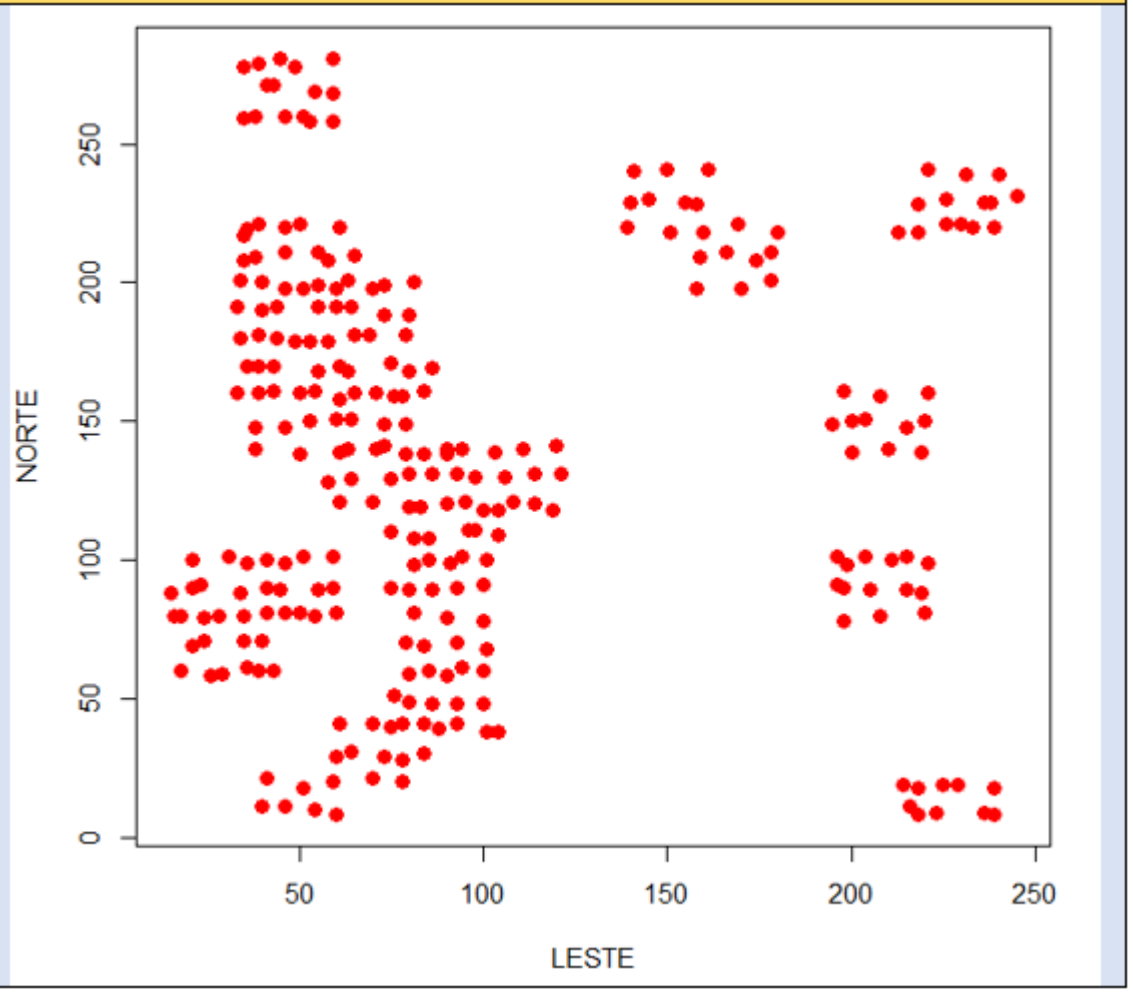
Script mapa_localizacao_U.R

```
1  #Script mapa_localizacao_U.R
2  def.par=par(no.readonly=TRUE)
3  setwd("C:\\IPT2021\\dados\\walker_data")
4  dados=read.csv("walker_dat.csv",sep=";",header=T)
5  x=dados$Xlocation[dados$U!=-99.00]
6  y=dados$Ylocation[dados$U!=-99.00]
7  plot(x,y,xlab="LESTE",ylab="NORTE",pch=21,
8  bg="red",cex=1.25,col="red")
9  par(def.par)
```

Comentários:

- Os objetos x e y recebem apenas os pontos que a variável U é diferente de -99 (linhas 5-6);
- Usamos o símbolo 21 com cor de fundo e contorno em vermelho (linha 8).

Dispositivo gráfico

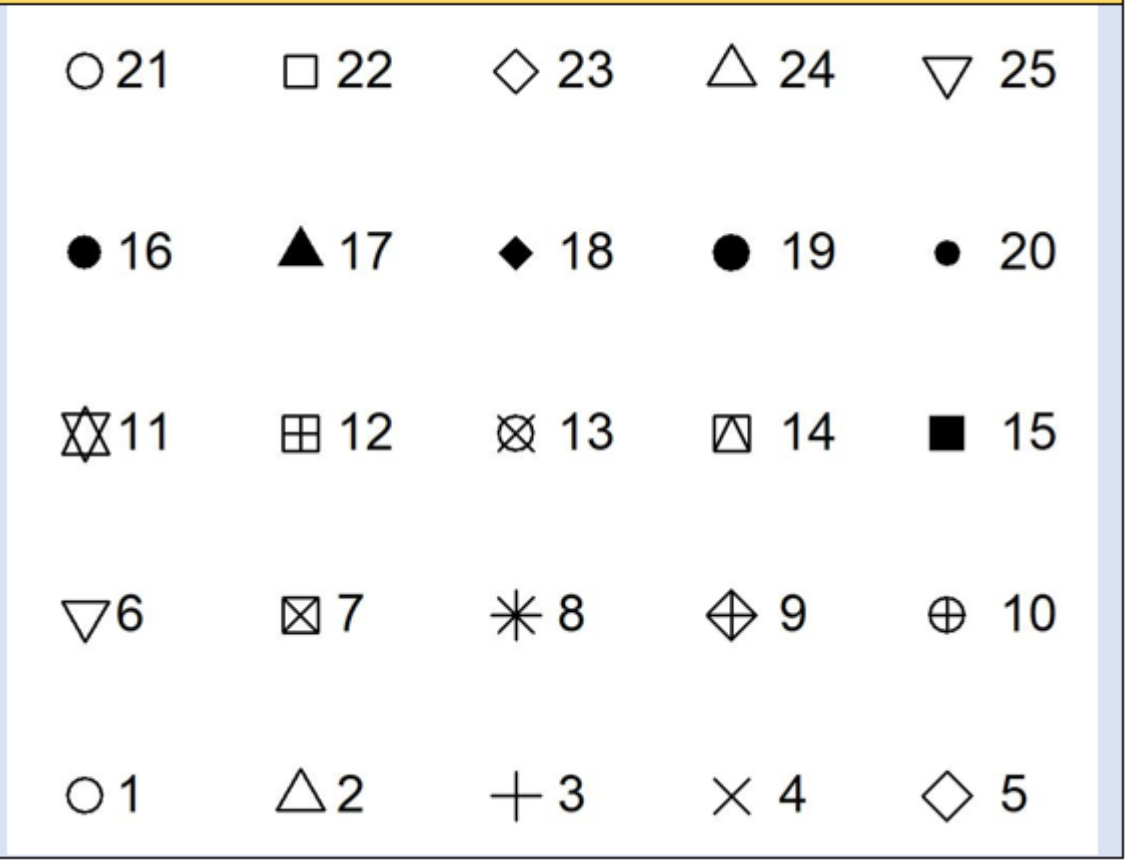


SÍMBOLOS PREDEFINIDOS PARA A FUNÇÃO PLOT()

Script simbolos.R

```
1  > #Script simbolos.R
2  > setwd("C:\\IPT2021\\figuras")
3  > jpeg("simbolos.jpeg",width=5,height=5,
4  + units="in",res=300)
5  > plot.new()
6  > for (i in 1:5){
7  +   for (j in 1:5){
8  +     kb=j+(i-1)*5
9  +     xp=(j-1)*0.20
10 +     yp=(i-1)*0.20
11 +     points(xp,yp,pch=kb,cex=1.5)
12 +     text((xp+0.025*xp),yp,pos=4,cex=1,
13 +     labels=as.character(kb))}
14 + }
15 > dev.off()
16 windows
17       2
18 >
```

Dispositivo gráfico



Os símbolos phc=21-25 são vazados e assim podem ser preenchidos com as cores de fundo (bg) e de contorno (col).
Veja o comando jpeg() que direciona a saída gráfica para o dispositivo jpeg.

Análise estatística

Distribuição de frequências

Uma lista desorganizada de números representando as realizações de experimentos não é facilmente assimilada (Benjamin e Cornell, 1970, p. 4). A simples tabulação dos dados brutos em uma distribuição de frequências permite obter uma primeira aproximação para verificação das características dos dados.

$$dados \rightarrow distr.freq. \rightarrow \begin{cases} descrição\ qualitativa \\ descrição\ quantitativa \end{cases}$$

No estudo da distribuição de frequências, assume-se:

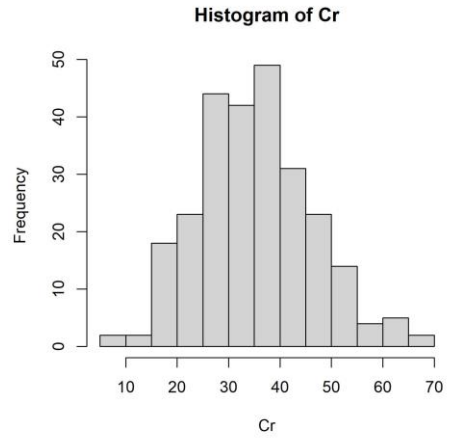
$$P(A_k) = \frac{1}{n}, k = 1,2, \dots, N$$

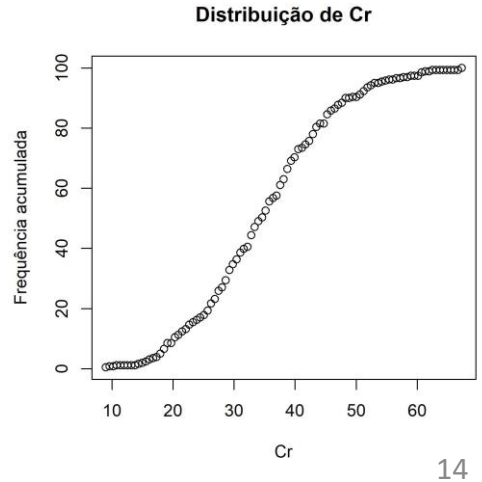
GEO KRIGAGEM

Análise estatística

Distribuição de frequências: descrição qualitativa

$$descrição\ qualitativa \rightarrow \begin{cases} histograma \\ curva\ acumulativa \end{cases}$$





Análise estatística

Distribuição de frequências: descrição qualitativa

O histograma é a representação gráfica da distribuição de frequências obtida agrupando os dados em classes.

- O tamanho da classe é calculado como:

$$xc = \frac{x_{max} - x_{min}}{nc}$$

- O número de classes pode ser determinado pela Regra de Sturges (Haan, 1977, p. 17-18):

$$nc = 1 + 3{,}222\log(\mathrm{n})$$

GEO KRIGAGEM

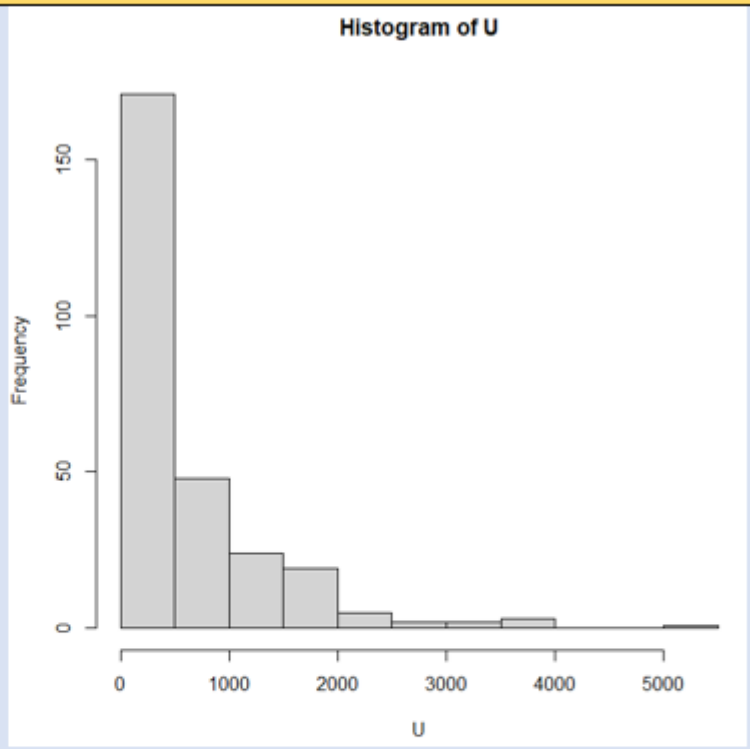
OBTENDO O HISTOGRAMA DA VARIÁVEL U

Script histograma_U.R

```
> #Script histograma_U.R
> setwd("C:\\IPT2021\\dados\\walker_data")
> dados=read.csv("walker_dat.csv",sep=";",header=T)
> U=dados$U[dados$U!=-99.00];
> h=hist(U)
> print(h)
$breaks
 [1]    0  500 1000 1500 2000 2500 3000 3500 4000 4500 5000 5500

$counts
 [1] 171  48  24  19   5   2   2   3   0   0   1

$density
 [1] 1.243636e-03 3.490909e-04 1.745455e-04 1.381818e-04 3.636364e-05
 [6] 1.454545e-05 1.454545e-05 2.181818e-05 0.000000e+00 0.000000e+00
[11] 7.272727e-06

$mids
 [1]  250  750 1250 1750 2250 2750 3250 3750 4250 4750 5250

$xname
[1] "U"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
>
```

Este é um exemplo de aplicação de uma função de alto nível. O histograma é obtido sem nenhuma interferência do usuário. Veja os componentes da função hist().

Dispositivo gráfico



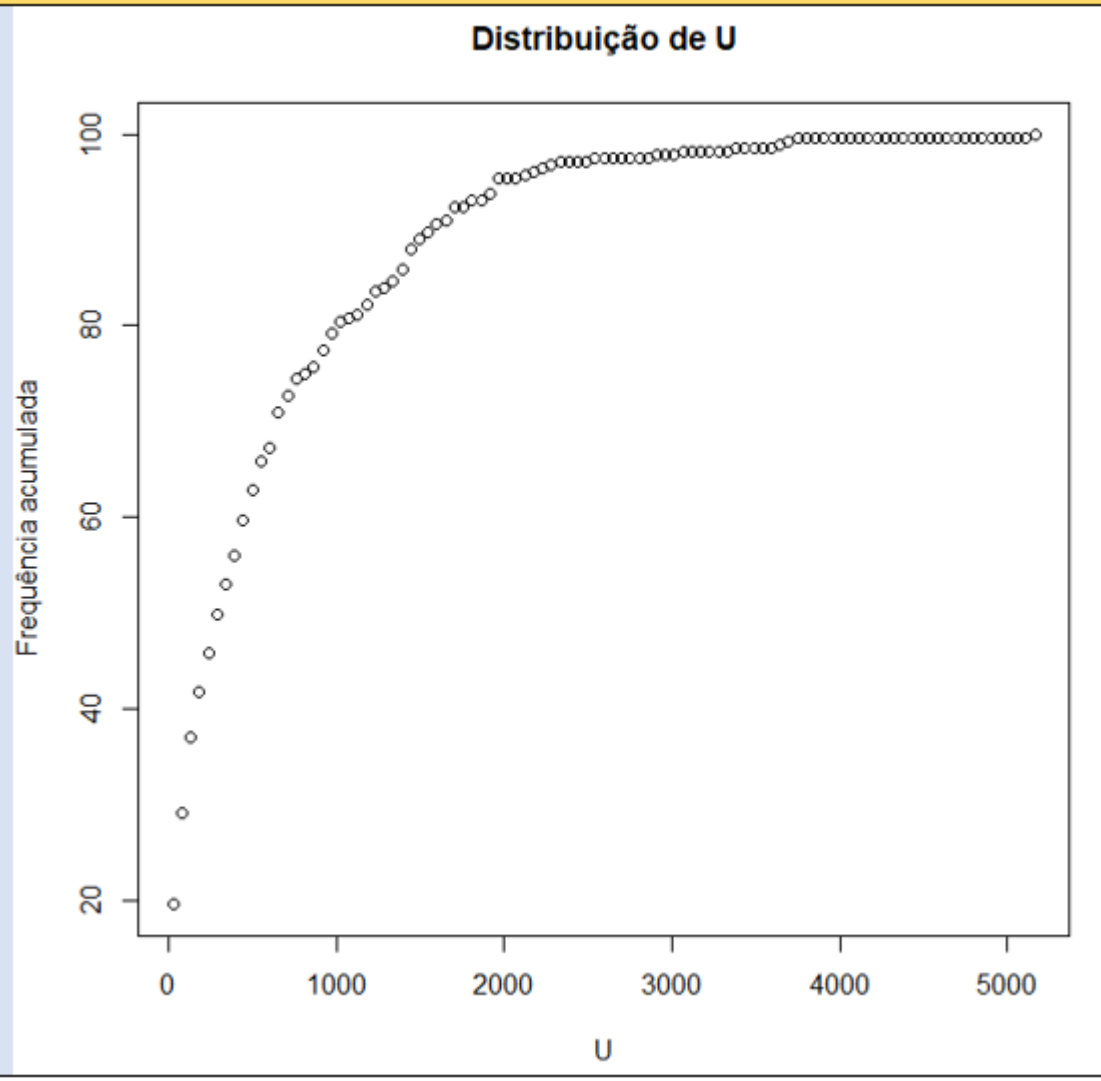
OBTENDO A CURVA ACUMULATIVA DA VARIÁVEL U

Script acumulativa_U.R

```
1  #Script acumulativa_U.R
2  quebras=seq(min(U),max(U),length=100)
3  h=hist(U,breaks=quebras,plot=F)
4  freq_acum=100*cumsum(h$counts)/sum(h$counts)
5  plot(h$mids,freq_acum,xlab="U",
6  ylab="Frequência acumulada",
7  main="Distribuição de U")
```

Aqui, usamos a componente counts da função hist(). O operador $ é aplicado para recuperar a componente do objeto h. Observe na linha 4 a natureza orientada a objetos da linguagem R.

Dispositivo gráfico



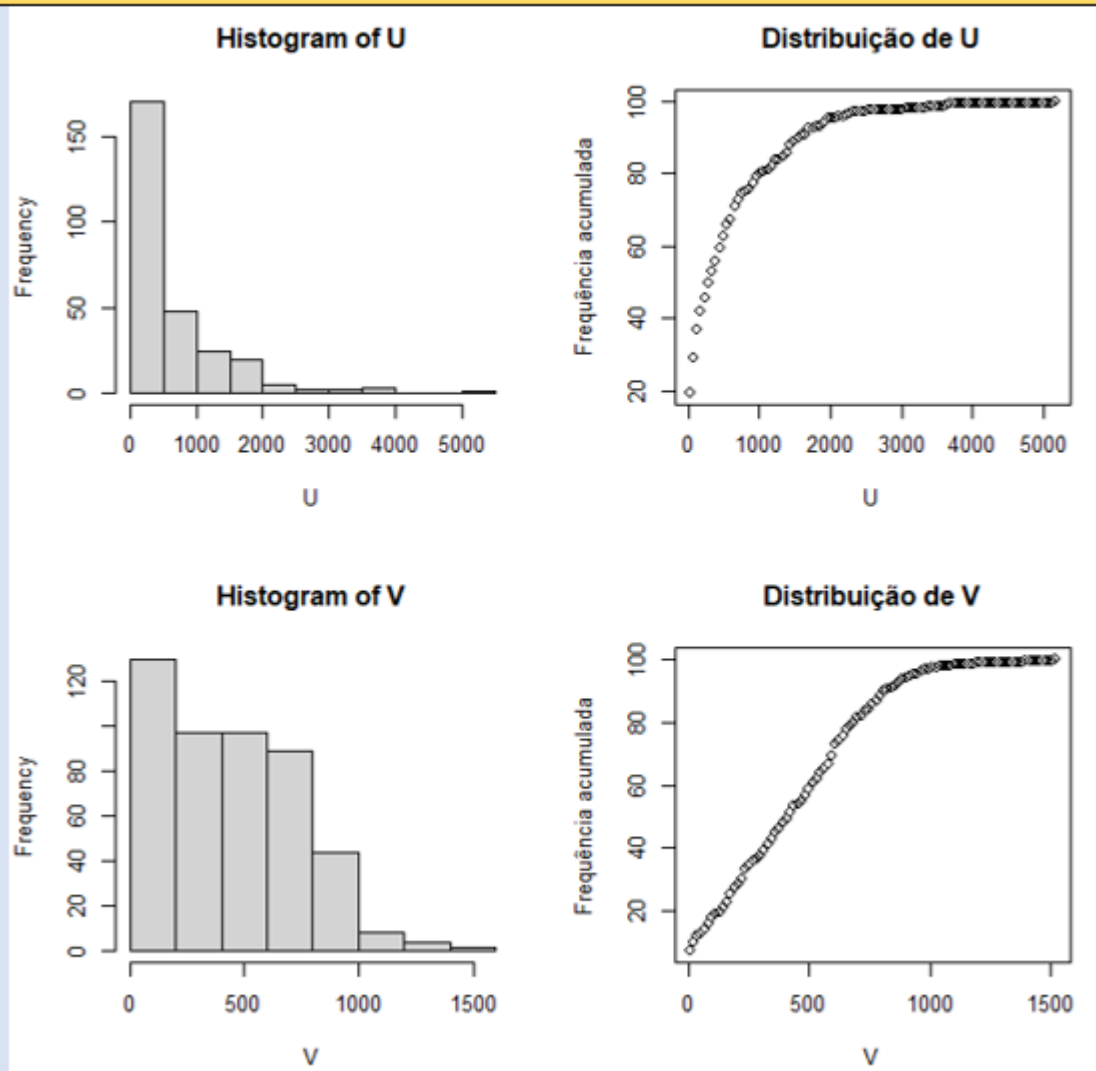
DESENHANDO HISTOGRAMAS E CURVAS ACUMULATIVAS PARA U E V

Script histo_acumulativas.R

```
1   #Script histo_acumulativas.R
2   par(mfrow=c(2,2))
3   hist(U)
4   quebras=seq(min(U),max(U),length=100)
5   h=hist(U,breaks=quebras,plot=F)
6   freq_acum=100*cumsum(h$counts)/sum(h$counts)
7   plot(h$mids,freq_acum,xlab="U",
8   ylab="Frequência acumulada",
9   main="Distribuição de U")
10  hist(V)
11  quebras=seq(min(V),max(V),length=100)
12  h=hist(V,breaks=quebras,plot=F)
13  freq_acum=100*cumsum(h$counts)/sum(h$counts)
14  plot(h$mids,freq_acum,xlab="V",
15  ylab="Frequência acumulada",
16  main="Distribuição de V")
17  par(def.par)
```

Na linha 2, definimos uma matriz de 2 X 2. Os gráficos serão plotados na sequência e por linha: histograma de U, curva acumulativa de U, histograma de V e curva acumulativa de V. A linha 17 retorna aos parâmetros originais.

Dispositivo gráfico



Análise estatística

Distribuição de frequências: descrição quantitativa

$$descr.quantitativa \rightarrow estatísticas\ descritivas \begin{cases} tendência\ central \rightarrow \begin{cases} média \\ mediana \\ moda \end{cases} \\ dispersão \begin{cases} variância \\ desvio\ padrão \\ coef.de\ variação \end{cases} \\ forma \begin{cases} assimetria \\ curtose \end{cases} \end{cases}$$

Análise estatística

Medidas de tendência central: média

Média ou esperança matemática:

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{N} x_i$$

Propriedade	Descrição
$E[K] = K$	A média de uma constante é a própria constante;
$E[KX] = KE[X]$	A média de uma variável multiplicada por uma constante é igual a constante vezes a média;
$E[X \pm Y] = E[X] \pm E[Y]$	A média da soma ou subtração de duas variáveis aleatórias é igual à soma ou subtração das médias;
$E[X \pm K] = E[X] \pm K$	A soma ou subtração de uma constante à variável aleatória é igual à média mais ou menos a constante.

Análise estatística

Medidas de tendência central: mediana

$$mediana \begin{cases} se\ n\ for\ ímpar \rightarrow X_{50} = x_{\frac{n+1}{2}} \\ se\ n\ for\ par \rightarrow X_{50} = \dfrac{\left(x_{n/2} + x_{n/2+1}\right)}{2} \end{cases}$$

GEO KRIGAGEM

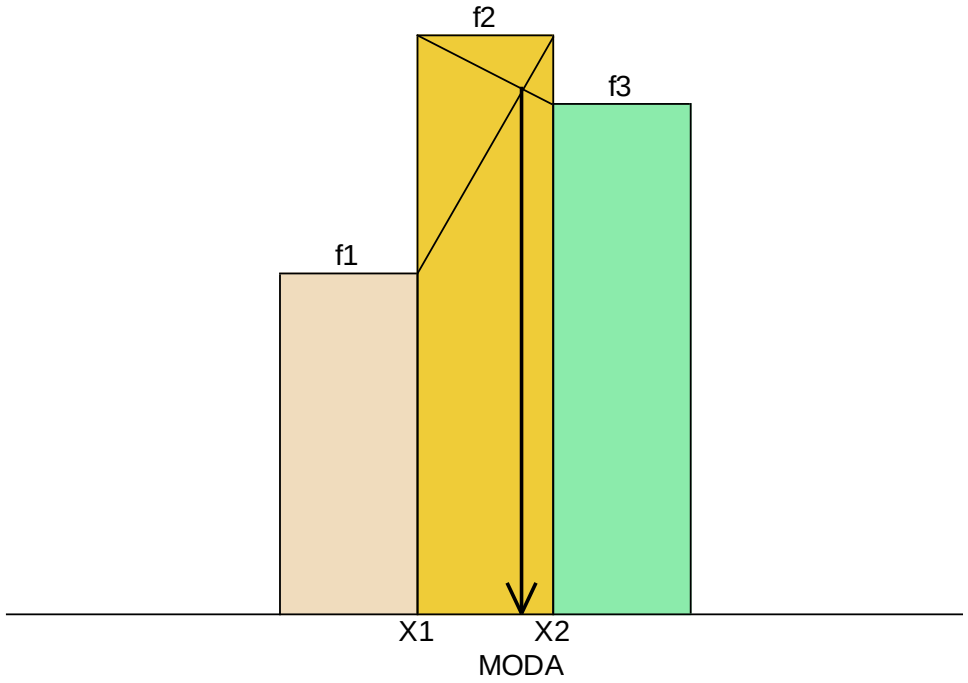
Análise estatística

Medidas de tendência central: moda

A interpolação da moda pode ser feita com base na seguinte fórmula (Francis, 2004, p. 118):

$$Moda = L + \frac{D_1}{D_1 + D_2} C$$

onde L é o limite inferior da classe modal; C é a largura da classe modal; D_1 é a diferença entre a maior frequência e a frequência da classe imediatamente anterior; D_2 é a diferença entre a maior frequência e frequência da classe imediatamente seguinte.



Fonte: Yamamoto (2020, p. 57)

Análise estatística

Função moda_francis()

```
moda_francis=function(x,nc){
  #quando a classe modal for 1 ou nc
  #a moda e o ponto medio
  #altere o nc para melhor definicao
  quebras=c(rep(0,(nc+1)))
  zc=(max(x)-min(x))/nc
  for (i in 1:(nc+1)){
  quebras[i]=min(x)+(i-1)*zc}
  h=hist(x,breaks=quebras,plot=FALSE)
  contagens=h$counts
  maximo<-max(contagens)
  indice<-which.max(contagens)
  largura<-quebras[2]-quebras[1]
  if (indice==1){
  d1=1; d2=1} else {
  if (indice==nc){
  d1=1; d2=1}
  else {
    d1<-contagens[indice]-contagens[indice-1]
    d2<-contagens[indice]-contagens[indice+1]}
  }
  moda<-quebras[indice]+(d1/(d1+d2))*largura
  return(moda)
}
```

Comentários:

- Calculamos as quebras do histograma para as nc classes;
- Usamos a função hist() para fazer as contagens nas classes, com a opção plot=FALSE;
- Localizamos a classe modal (maior contagem);
- Calculamos a largura que é o tamanho da classe;
- Se a classe modal for a primeira (1) ou a última (nc), a moda será o ponto médio;
- Caso contrário, aplica-se a fórmula de Francis (2014, p. 118)
- Esta função está no script estatisticas.R

Análise estatística

Medidas de dispersão: variância

Variância:

$$S^2 = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \overline{X}\right)^2$$

Propriedade	Descrição
$Var[K] = 0$	Constante não tem variância;
$Var[KX] = K^2 Var[X]$	A variância de uma constante vezes X é igual à essa constante ao quadrado vezes a variância de X;
$Var[X \pm Y] = Var[X] + Var[Y] \pm 2Cov(X, Y)$	A variância da soma ou subtração de duas variáveis é a soma das variâncias mais ou menos duas vezes a covariância entre elas;
$Var[X \pm K] = Var[X]$	A soma ou subtração de uma constante não altera a variância da variável aleatória.

Análise estatística

Medidas de dispersão: desvio padrão e coeficiente de variação

Desvio padrão:

$$S = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \overline{X}\right)^2}$$

Coeficiente de variação:

$$CV = \frac{S}{\overline{X}}$$

GEO KRIGAGEM

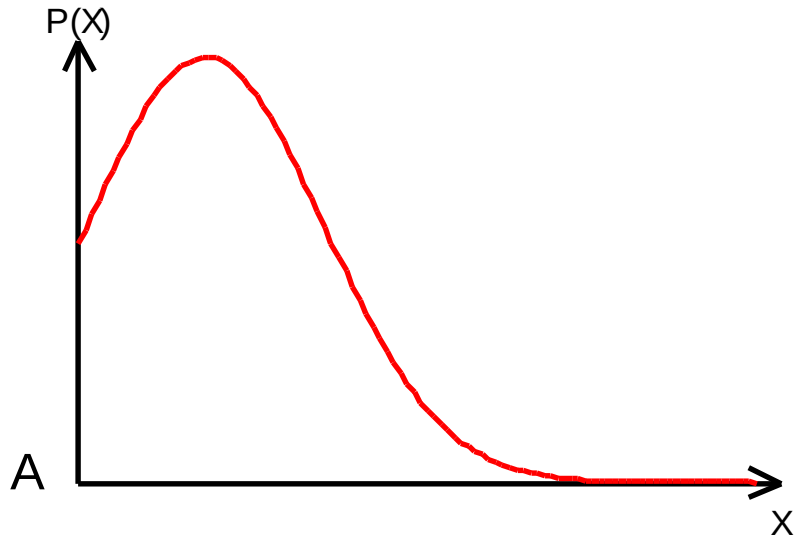
Análise estatística

Medidas de forma: assimetria

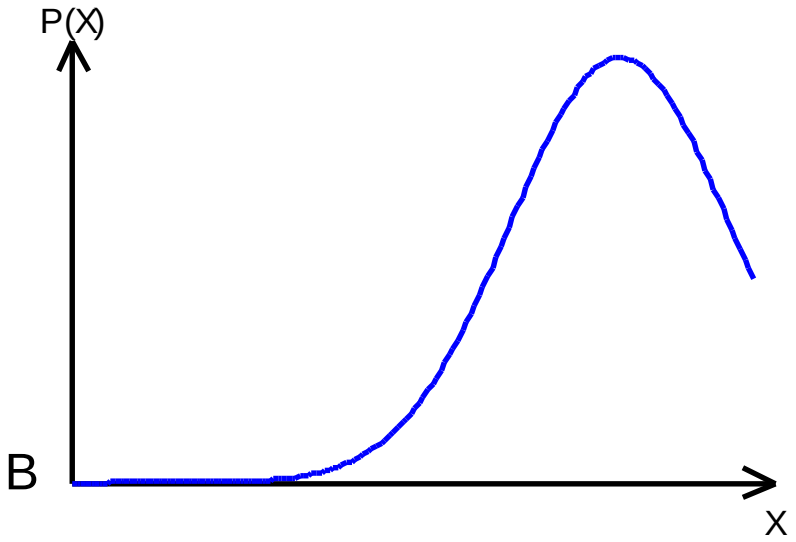
Coeficiente de assimetria:

$$CA = \frac{1}{n}\sum_{i=1}^{n}\frac{\left(x_i - \overline{X}\right)^3}{S^3}$$

CA > 0: assimetria positiva



CA < 0: assimetria negativa



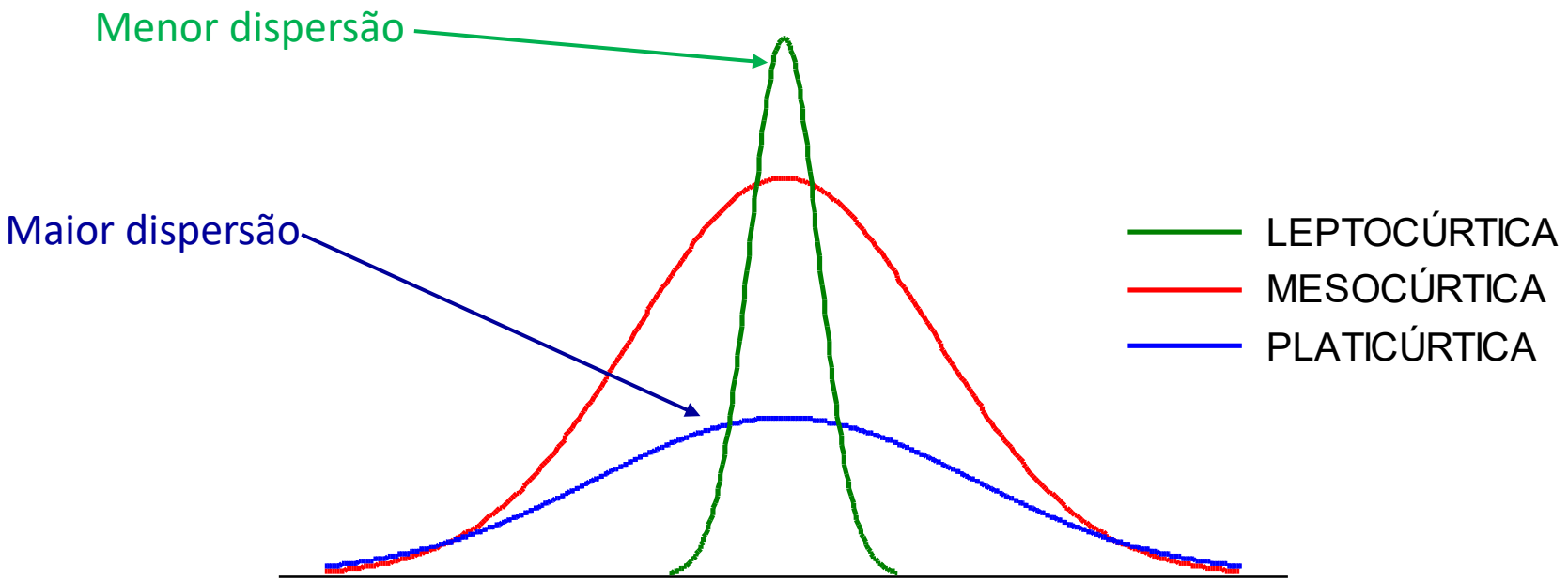
Fonte: Yamamoto (2020, p. 63)

Análise estatística

Medidas de forma: curtose

Coeficiente de curtose:

$$CC = \frac{1}{n}\sum_{i=1}^{n}\frac{\left(x_i - \overline{X}\right)^4}{S^4}$$



Fonte: Yamamoto (2020, p. 63)

Análise estatística

Divisores da distribuição de frequências

$$divisores \begin{cases} quartis \rightarrow quatro\ partes: 25, 50\ e\ 75\% \\ decis \rightarrow dez\ partes: 10, 20, 30, \ldots, 70, 80\ e\ 90\% \\ percentis \rightarrow cem\ partes: 1, 2, \ldots, 98\ e\ 99\% \end{cases}$$

A mediana é o segundo quartil ou quinto decil ou quinquagésimo percentil.
A amplitude interquartil (medida de dispersão) é calculada como:

$$AIQ = Q_3 - Q_1$$

Função descritivas()

```
descritivas=function(x){
  library(moments)
  n=length(x)
  media=mean(x)
  x50=median(x)
  vmin=min(x)
  vmax=max(x)
  S2=var(x)
  S=sd(x)
  cv=S/media
  moda=moda_francis(x,12)
  print(moda)
  ass=skewness(x)
  kur=kurtosis(x)
  qua=quantile(x)
  aiq=IQR(x)
  res=c(n,media,x50,moda,S2,S,cv,ass,
  kur,qua,aiq,vmin,vmax)
  return(res)
}
```

Comentários:

- Esta função precisa do pacote “moments” para cálculo da assimetria e curtose;
- Chama a função moda_francis() descrita anteriormente;
- A função quantile() retorna os cinco quantis (0, 25, 50, 75 e 100%);
- Portanto o vetor res tem tamanho 17.

Função descritivas() para as estatísticas descritivas de V

Script estatisticas.R

```
1  setwd("C:\\IPT2021\\dados\\walker_data")
2  dados=read.csv("walker_dat.csv",sep=";",header=T)
3  V=dados$V[dados$V!=-99.00];
4  #estatisticas para V
5  res=round(descritivas(V),digits=3)
6  estats=c(
7  "No. de dados","Média","Mediana",
8  "Moda","Variância","Desvio Padrão",
9  "Coef. de variação",
10 "Amplitude interquartil",
11 "Assimetria","Curtose","Valor mínimo",
12 "Quartil Inferior","Quartil Superior",
13 "Valor máximo")
14 valores=c(round(res[1],digits=0),
15 res[2],res[3],res[4],res[5],
16 res[6],res[7],res[15],res[8],
17 res[9],res[16],res[11],res[13],
18 res[17])
19 print(cbind(estats,valores))
```

Console

```
     estats                    valores
     "No. de dados"            "470"
     "Média"                   "435.299"
     "Mediana"                 "424"
     "Moda"                    "63.671"
     "Variância"               "89929.395"
     "Desvio Padrão"           "299.882"
     "Coef. de variação"       "0.689"
     "Amplitude interquartil" "456.25"
     "Assimetria"              "0.459"
     "Curtose"                 "2.871"
     "Valor mínimo"            "0"
25%  "Quartil Inferior"        "184.6"
75%  "Quartil Superior"        "640.85"
     "Valor máximo"            "1528.1"
```

Análise estatística

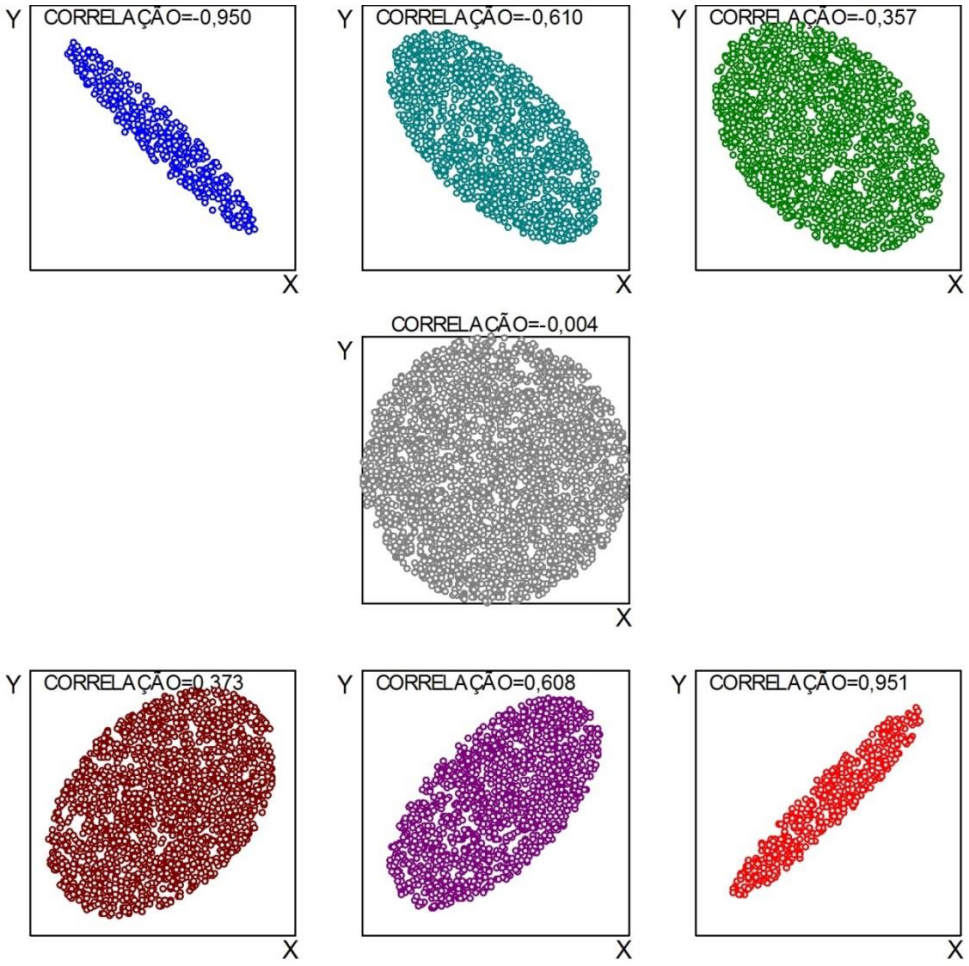
Medidas de relação mútua entre duas variáveis aleatórias

As relações mútuas entre duas variáveis aleatórias podem ser determinadas se calculando a covariância e o coeficiente de correlação, conforme as fórmulas:

$$Cov(x,y) = \frac{1}{(n-1)} \sum_{i=1}^{n} (x_i - \overline{x})\,(y_i - \overline{y})$$

e

$$\rho_{x,y} = \frac{Cov(x,y)}{S_x S_y}$$



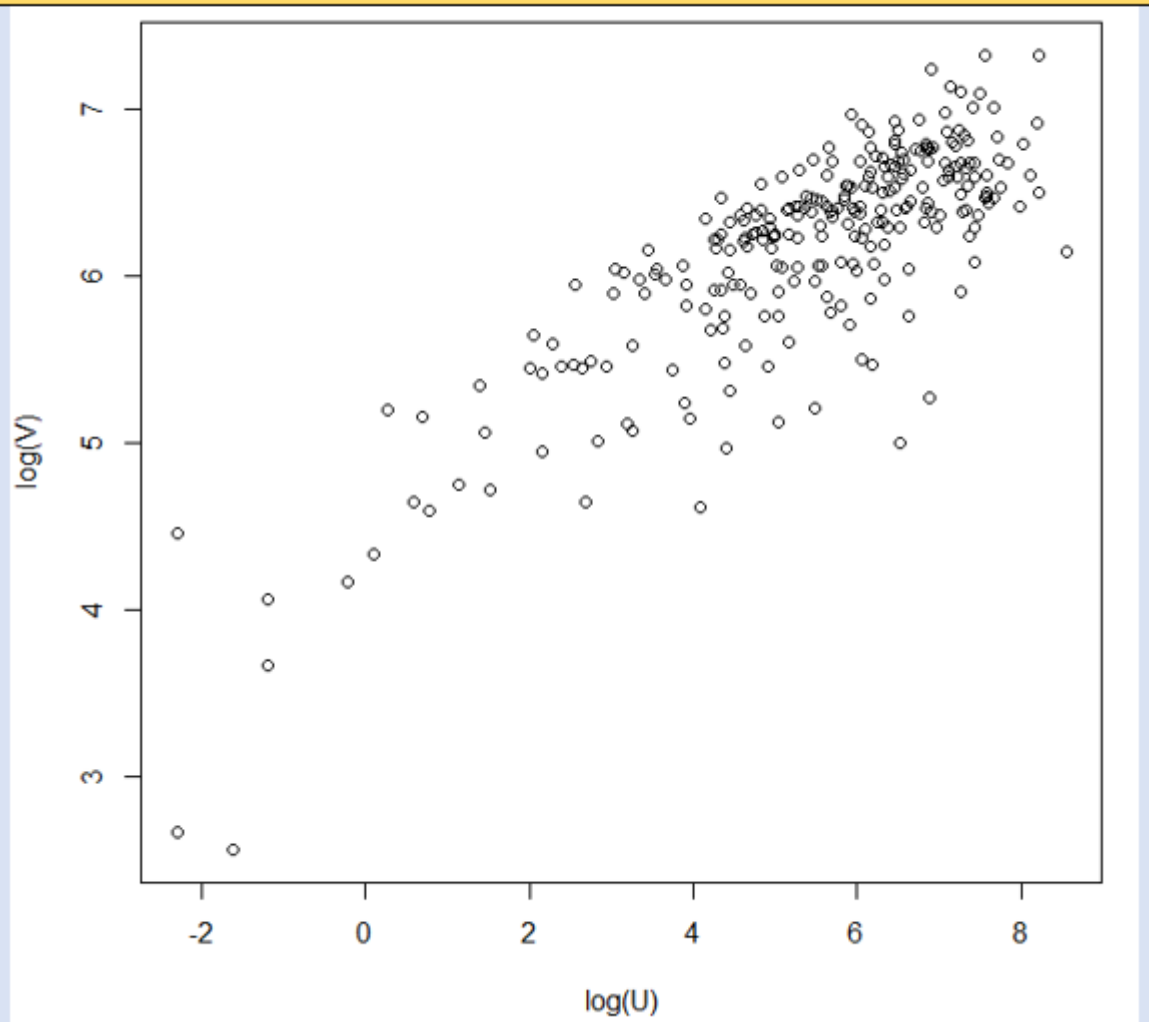
Fonte: Yamamoto (2020, p.73)

Relações mútuas entre U e V

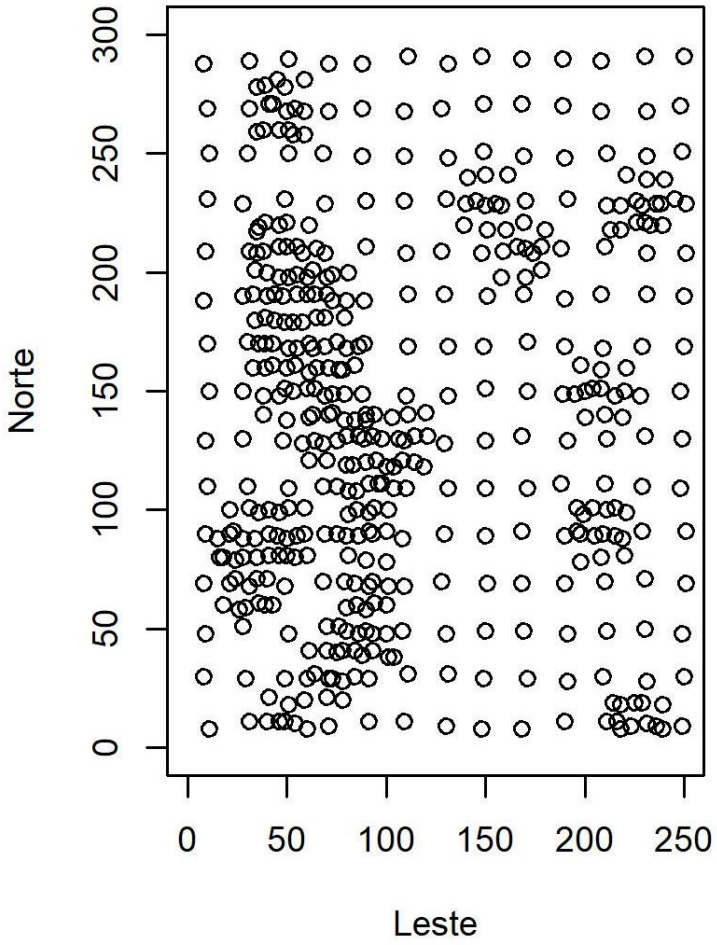
Script covarCor.R

```
> plot(xc,yc)
> #Script covarCor.R
> setwd("C:\\IPT2021\\dados\\walker_data")
>
dados=read.csv("walker_dat.csv",sep=";",header=T)
> x=dados$U; y=dados$V
> n=length(x)
> xc=c(rep(0,n)); yc=c(rep(0,n))
> code=-99.00; k=0
> #codigo -99
> for (i in 1:n){
+   if(x[i] != code & y[i] != code){
+   k=k+1
+   xc[k]=x[i]
+   yc[k]=y[i]}}
> print(k)
[1] 275
> plot(xc,yc)
> print(c(cov(x,y),cor(x,y)))
[1] 1.160065e+05 5.677105e-01
>
plot(log(xc),log(yc),xlab="log(U)",ylab="log(V)")
>
```

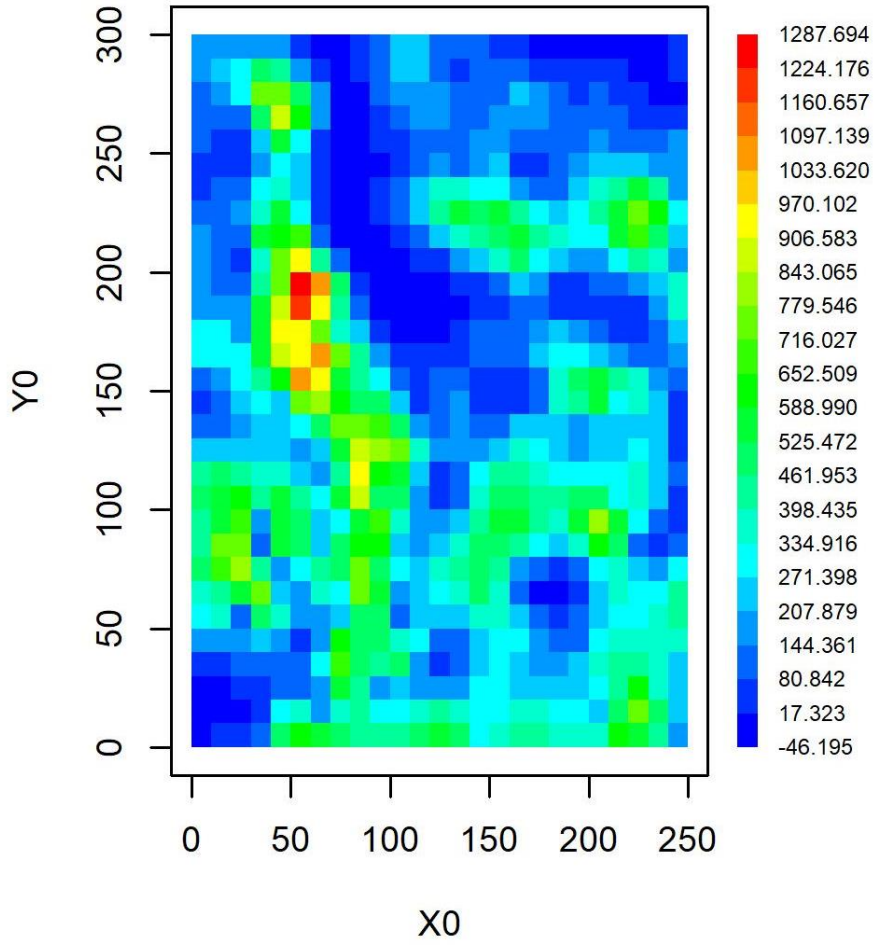
Dispositivo gráfico



COMO TRANSFORMAR PONTOS EM MAPA DA DISTRIBUIÇÃO ESPACIAL

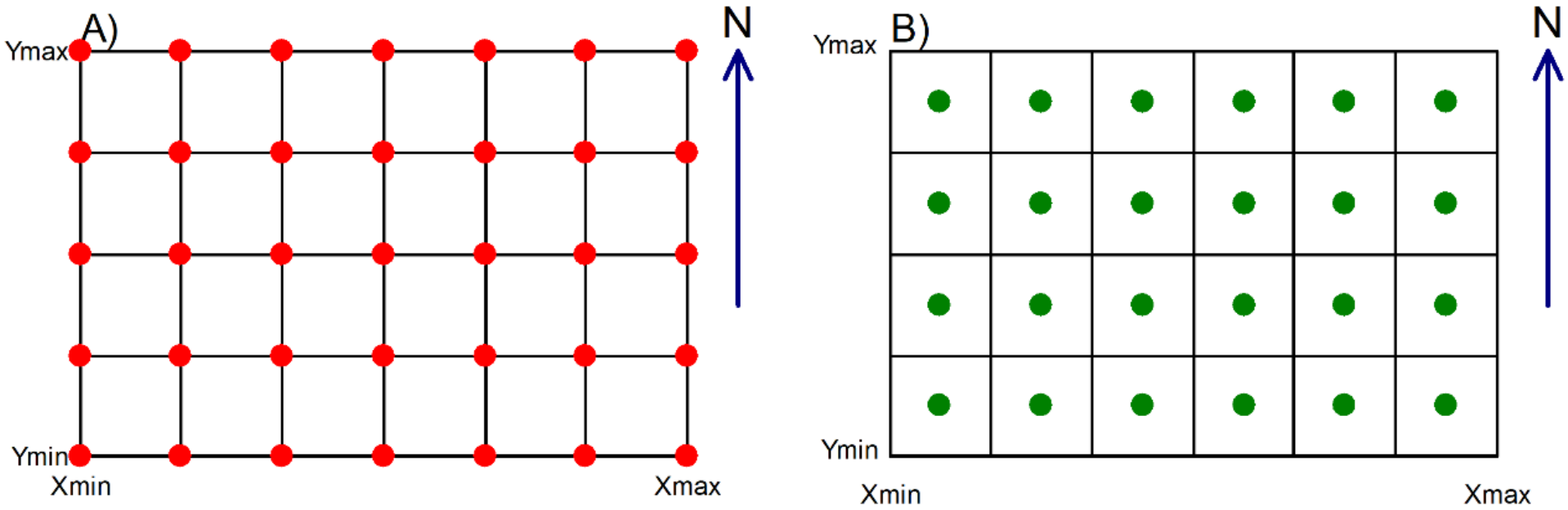


interpolação



Interpolação de dados 2D

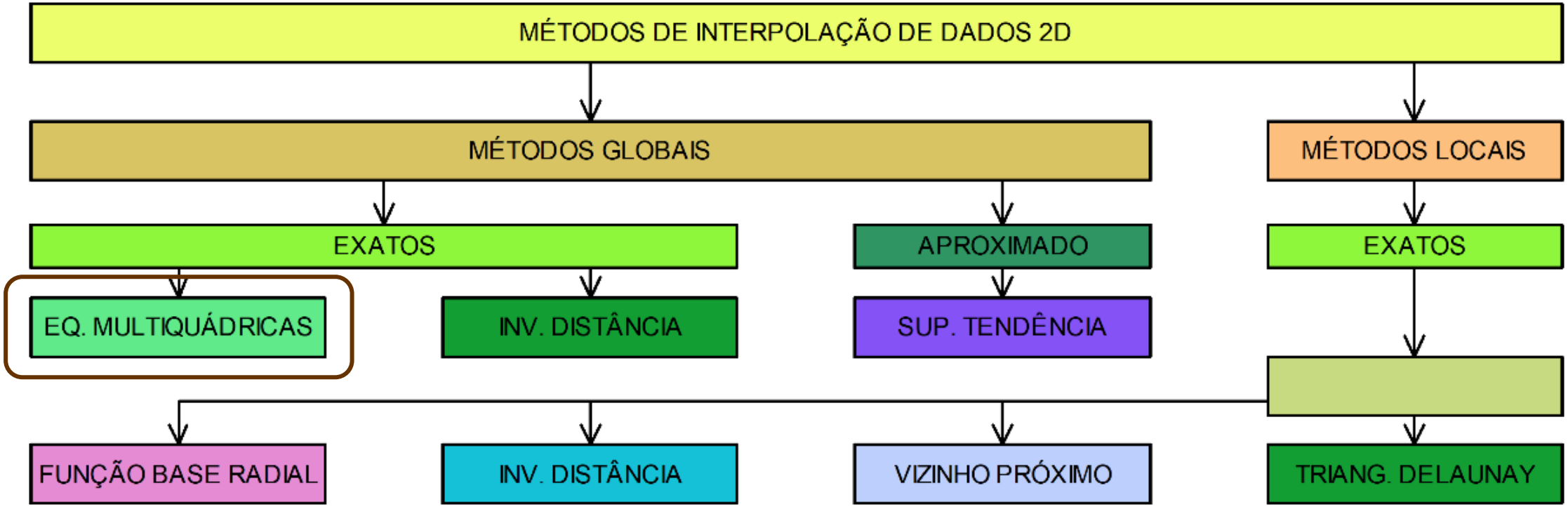
Tipos de malha regular: A) sobre os nós; B) nos centros



Fonte: Yamamoto (2020, p. 143).

Interpolação de dados 2D

Métodos de interpolação de dados 2D



Fonte: Yamamoto (2020, p. 148).

Interpolação de dados 2D

Equações multiquádricas globais (Hardy, 1971)

A forma geral da equação multiquádrica em 2D, segundo Hardy (1971, p. 1906):

$$Z^*(\boldsymbol{u_o}) = \sum_{i=1}^{N} c_i \left[(x_i - x_o)^2 + (y_i - y_o)^2 + C\right]^{1/2}$$

Onde os coeficientes $\{c_i, i = 1, N\}$ são determinados pela resolução de um sistema de equações lineares (Hardy, 1971, p. 1907):

$$Z(\boldsymbol{u}_i) = \sum_{j=1}^{N} c_i \left[\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + C\right]^{1/2} \quad \text{para i=1,N}$$

GEO KRIGAGEM

Interpolação de dados 2D

Equações multiquádricas globais (Hardy, 1971)

$$\boldsymbol{Qc = z}$$

Em forma matricial, tem-se:

$$\begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1N} \\ q_{21} & q_{22} & \cdots & q_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N1} & q_{N2} & \cdots & q_{NN} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} Z(u_1) \\ Z(u_2) \\ \vdots \\ Z(u_n) \end{bmatrix}$$

Onde $q_{ij} = \left[(x_i - x_j)^2 + (y_i - y_j)^2 + C \right]^{1/2}$

C é uma constante positiva e, segundo Franke (1982, p. 191), o método é bastante estável com relação à essa constante dando, consistentemente, bons resultados.

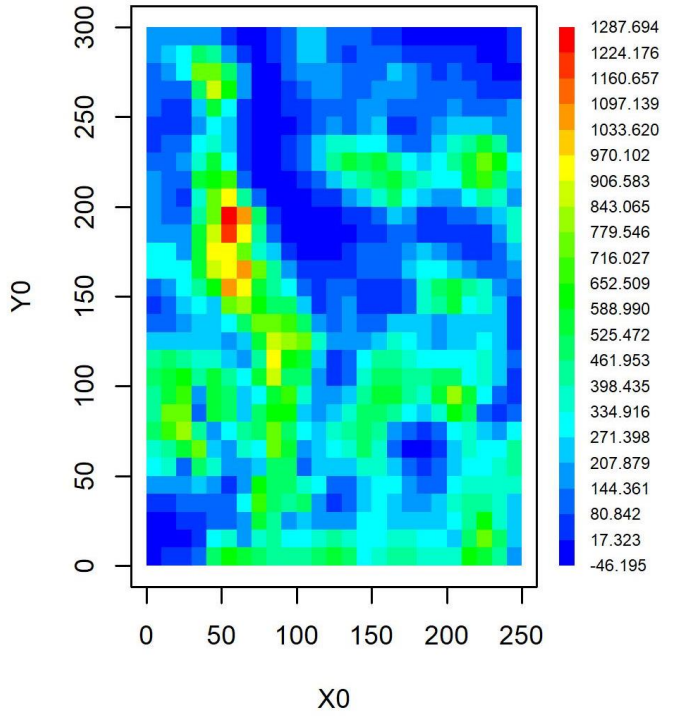
GEO KRIGAGEM

DADOS DE ENTRADA

Arquivo de dados: walker_dat.csv

1	ID	Xlocation	Ylocation	V	U	T
2	1	11	8	0	-99	2
3	2	8	30	0	-99	2
4	3	9	48	224.4	-99	2
5	4	8	69	434.4	-99	2
6	5	9	90	412.1	-99	2
7	6	10	110	587.2	-99	2
8	7	9	129	192.3	-99	2
9	8	11	150	31.3	-99	2
10	9	10	170	388.5	-99	2
11	10	8	188	174.6	-99	2
12	11	9	209	187.8	-99	2
13	12	10	231	82.1	-99	1
14	13	11	250	81.1	-99	1
15	14	10	269	124.3	-99	2

Como fazer esse mapa em 10 passos!



Arquivo de parâmetros: walker_dat_par.csv

1	xmin	xmax	dx	nx	ymin	ymax	dy	ny	code
2	0	250	10	25	0	300	10	30	-999

LEITURA DOS DADOS DE ENTRADA

1

Script – leitura dos parametros e dados

```
1   > #script emqGlobal.R
2   > #escrito por Jorge Kazuo Yamamoto
3   > setwd("C:\\IPT2021\\dados\\walker_data")
4   > #script - leitura dos parametros e dados
5   > #leitura do arquivo de parametros
6   > par<- read.csv("walker_dat_par.csv",sep=";",header=TRUE)
7   > xmin=par$xmin; xmax=par$xmax; dx=par$dx; nx=par$nx
8   > ymin=par$ymin; ymax=par$ymax; dy=par$dy; ny=par$ny
9   > code=par$code  #valor nao calculado
10  > #leitura do arquivo de dados
11  > dados <- read.csv("walker_dat.csv",sep=";",header=TRUE)
12  > x=dados$Xlocation[dados$V!=-99.00]
13  > y=dados$Ylocation[dados$V!=-99.00]
14  > z=dados$V[dados$V!=-99.00]
15  > n=length(z)
16  > print(c(n,min(z),max(z)))
17  [1]  470.0    0.0 1528.1
18  >
```

O objeto dados é um data frame.

SISTEMA DE EQUAÇÕES MULTIQUÁDRICAS

2

Script – sistema de equacoes multiquadricas

```
1  #script sistema de equacoes multiquadricas
2  C=1  #constante multiquadrica
3  Q=matrix(c(rep(0,n*n)),nrow=n)
4  #calculando a matriz dos coeficientes
5  for (i in 1:n){
6    for (j in 1:n){
7      Q[i,j]=sqrt((x[i]-x[j])^2+(y[i]-y[j])^2+C)}}
8  z=matrix(c(z),ncol=1)
9  c=solve(Q,z)
```

$$\boldsymbol{Qc = z}$$

$$\begin{bmatrix} Z(u_1) \\ Z(u_2) \\ \vdots \\ Z(u_n) \end{bmatrix}$$

$$\begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1N} \\ q_{21} & q_{22} & \cdots & q_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N1} & q_{N2} & \cdots & q_{NN} \end{bmatrix}$$

INTERPOLAÇÃO DA MALHA REGULAR

3

Script – interpolacao da malha regular

```
1   #Script interpolacao da malha regular
2   print(c(nx,ny))
3   x0=c(rep(0,nx*ny)); y0=c(rep(0,nx*ny)); z0=c(rep(0,nx*ny))
4   for (i in 1:ny){
5     for (j in 1:nx){
6       kb=j+(i-1)*nx
7       x0[kb]=xmin+(j-1)*dx+dx/2
8       y0[kb]=ymin+(i-1)*dy+dy/2
9       z0[kb]=0
10      for (k in 1:n){
11        z0[kb]=z0[kb]+c[k]*sqrt((x[k]-x0[kb])^2+(y[k]-y0[kb])^2+C)}
12    }
13  }
```

$$Z^*(\boldsymbol{u_o}) = \sum_{i=1}^{N} c_i \left[(x_i - x_o)^2 + (y_i - y_o)^2 + C\right]^{1/2}$$

GRAVAÇÃO DA MALHA REGULAR EM ARQUIVO *.CSV

4

Script – gravacao arquivo csv da malha regular

```
1  #Script gravacao arquivo csv da malha regular
2  saida=data.frame(x0,y0,z0)
3  names(saida)[1]=paste("X0")
4  names(saida)[2]=paste("Y0")
5  names(saida)[3]=paste("Z0")
6  setwd("C:\\IPT2021\\dados\\walker_data")
7  write.csv(saida,file="walker_dat_EMQ.csv",row.names=FALSE,quote=F)
```

LEITURA DO ARQUIVO DE CORES RGB

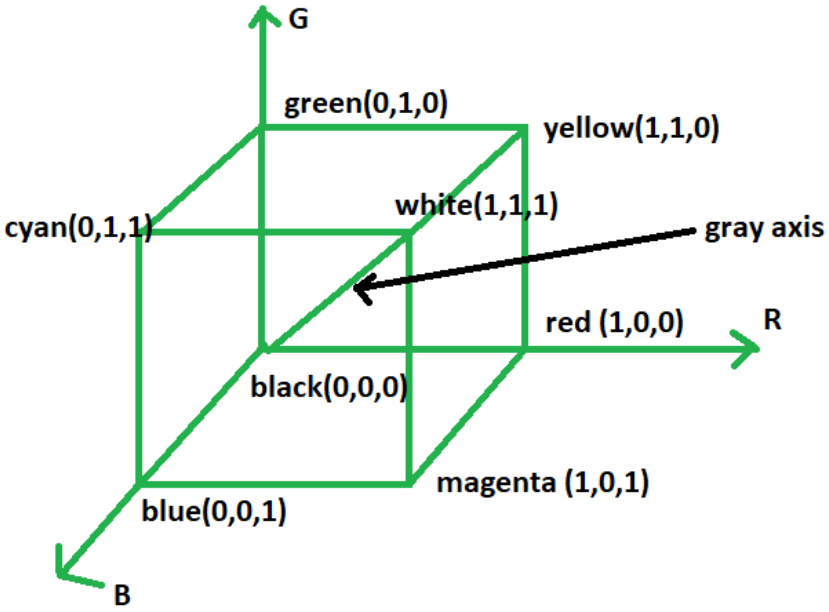
5

Para geração do mapa imagem – precisamos de um arquivo de cores RGB

Script – leitura do arquivo de cores

```
1  #Script leitura do arquivo de cores
2  def.par=par(no.readonly=TRUE)
3  setwd("C:\\IPT2021\\dados\\cores")
4  cores <- read.csv("cores.csv",sep=";",header=TRUE)
5  r=cores$r; g=cores$g; b=cores$b
6  ncores=length(r)
```

Cor	R	G	B	Cor	R	G	B
1	0,00	0,00	1,00	11	0,00	1,00	0,00
2	0,00	0,20	1,00	12	0,20	1,00	0,00
3	0,00	0,40	1,00	13	0,40	1,00	0,00
4	0,00	0,60	1,00	14	0,60	1,00	0,00
5	0,00	0,80	1,00	15	0,80	1,00	0,00
6	0,00	1,00	1,00	16	1,00	1,00	0,00
7	0,00	1,00	0,80	17	1,00	0,80	0,00
8	0,00	1,00	0,60	18	1,00	0,60	0,00
9	0,00	1,00	0,40	19	1,00	0,40	0,00
10	0,00	1,00	0,20	20	1,00	0,20	0,00
...	...	...	...	21	1,00	0,00	0,00



$$No.Cores = 2^{24} - 1 = 16{,}7\ mi$$

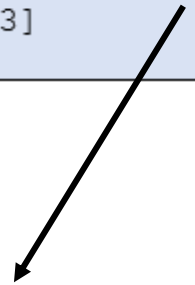
Fonte: https://www.geeksforgeeks.org/computer-graphics-the-rgb-color-model/

LEITURA DOS ARQUIVOS DE PARÂMETROS E DA MALHA REGULAR

6

Script – leitura dos arquivos: parametros e malha regular

```
1   #Script leitura dos arquivos: parametros e malha regular
2   setwd("C:\\IPT2021\\dados\\walker_data")
3   #leitura do arquivo de parametros
4   par<- read.csv("walker_dat_par.csv",sep=";",header=TRUE)
5   xmin=par$xmin; xmax=par$xmax; dx=par$dx; nx=par$nx
6   ymin=par$ymin; ymax=par$ymax; dy=par$dy; ny=par$ny
7   code=par$code  #valor nao calculado
8   #leitura do arquivo da malha regular *.csv
9   #o arquivo csv gerado pelo R tem a "," como separador
10  dados <- read.csv("walker_dat_EMQ.csv",sep=",",header=TRUE)
11  x=dados[,1]; y=dados[,2]; z=dados[,3]
12  n=length(z)
```



Agora, o separador é a vírgula, pois foi gravado pelo R.

DEFINIÇÃO DAS DIMENSÕES DO MAPA E DISPOSITIVO JPEG

7

Script – definicao da dimensao do mapa no dispositivo grafico novo

```
1  #Script - definicao da dimensao do mapa no dispositivo grafico novo
2  if ((xmax-xmin) > (ymax-ymin)){
3    cx=5
4    cy=(ymax-ymin)*cx/(xmax-xmin)
5  } else {
6    cy=5
7    cx=(xmax-xmin)*cy/(ymax-ymin)
8  }
9  dev.new(width=cx, height=cy, unit="in")
10 setwd("C:\\IPT2021\\figuras")
11 jpeg("walker_dat_map.jpeg",width=cx,height=cy,
12 units="in",res=300)
13 par(mar=c(5,5,10,10), xpd=TRUE,cex=0.75)
14 plot(NA,NA,xlim=c(xmin,xmax), ylim=c(ymin,ymax), type="n",
15 frame=TRUE, xlab=colnames(dados[1]), ylab=colnames(dados[2]))
```

Comentários:

- As dimensões cx e cy mantêm a proporcionalidade dos eixos;
- O comando dev.new() – linha 9 – define um novo dispositivo gráfico;
- As linhas 11-12 direcionam a saída gráfica para o dispositivo jpeg;
- As linhas 14-15 criam uma área de plotagem em branco com dimensões proporcionais a cx e cy.

DEFINIÇÃO DOS LIMITES DA VARIÁVEL

8

Script – definicao dos limites (zmin,zmax)

```
1   > #Script - definicao dos limites (zmin,zmax)
2   > zmin=min(z[z!=code]); zmax=max(z[z!=code])
3   > print(c(zmin,zmax))
4   [1]  -46.19505 1287.69434
5   > zmin=9.9e+20; zmax=-zmin
6   > for (i in 1:n){
7   +   if (z[i] != code)    {
8   +       if (z[i] < zmin) {zmin=z[i]}
9   +       if (z[i] > zmax) {zmax=z[i]}
10  +     }
11  + }
12  > print(c(zmin,zmax))
13  [1]  -46.19505 1287.69434
```

Comentários:

- Linha 2 aproveita a natureza orientada a objetos do R;
- Linhas 5-11 obtêm os mesmos resultados, mas por programação.

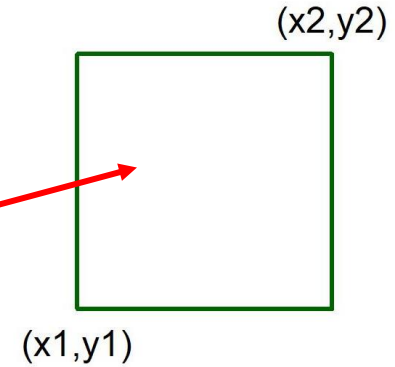
PLOTAGEM DOS RETÂNGULOS

9

Script – plotagem dos retangulos

```
1   #Script - plotagem dos retangulos
2   delta=(zmax-zmin)*1.0000001
3   #plotagem como retangulos
4   for (i in 1:n)  {
5     if (z[i] != code) {
6       x1=x[i]-dx/2
7       y1=y[i]-dy/2
8       x2=x[i]+dx/2
9       y2=y[i]+dy/2
10      cor=trunc((z[i]-zmin)*ncores/delta)+1
11      rect(x1,y1,x2,y2,border=NA,col=rgb(r[cor],g[cor],b[cor]))
12    }
13  }
```

Este retângulo recebe a cor RGB

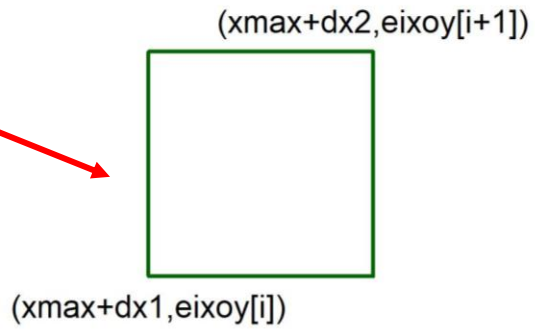


O retângulo é definido pelas coordenadas do canto inferior esquerdo e canto superior direito.

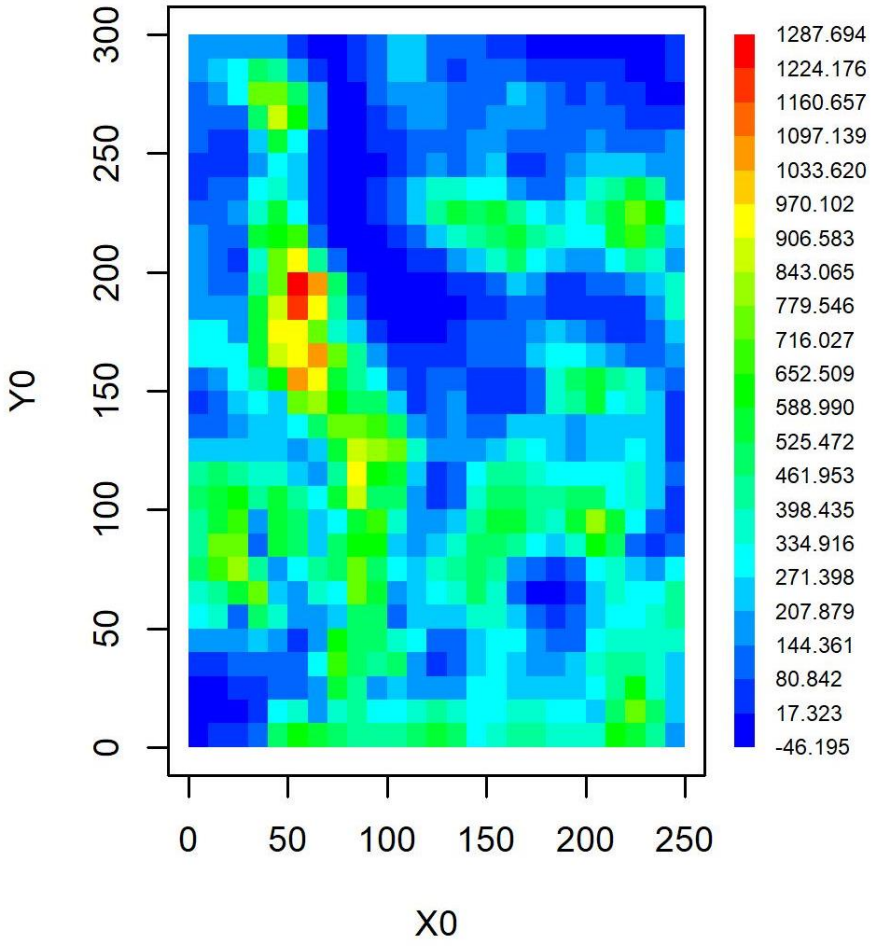
10 PLOTAGEM DA LEGENDA DE CORES

Script – plotagem da legenda de cores

```
1   #Script - plotagem da legenda de cores
2   dx1=(xmax-xmin)*0.10; dx2=(xmax-xmin)*0.14
3   eixox=c(rep(xmax+dx2,ncores+1))
4   eixoy=c(rep(0,ncores+1))
5   deltay=(ymax-ymin)/ncores
6   for (i in 1:(ncores+1))
7   {eixoy[i]=ymin+(i-1)*deltay}
8   eixoz=c(rep(0,ncores+1))
9   deltaz=(zmax-zmin)/ncores
10  for (i in 1:(ncores+1))
11  {eixoz[i]=zmin+(i-1)*deltaz}
12  vetorz=c(rep(0,ncores+1))
13  vetorz=sprintf("%.3f",eixoz)
14  text(x=c(eixox),y=c(eixoy),c(vetorz),xpd=NA,cex=0.65, pos=4)
15  for (i in 1:ncores)
16  {rect(xmax+dx1,eixoy[i],xmax+dx2,eixoy[i+1],border=NA,col=rgb(r[i],g[i],b[i]))}
17  dev.off()
```

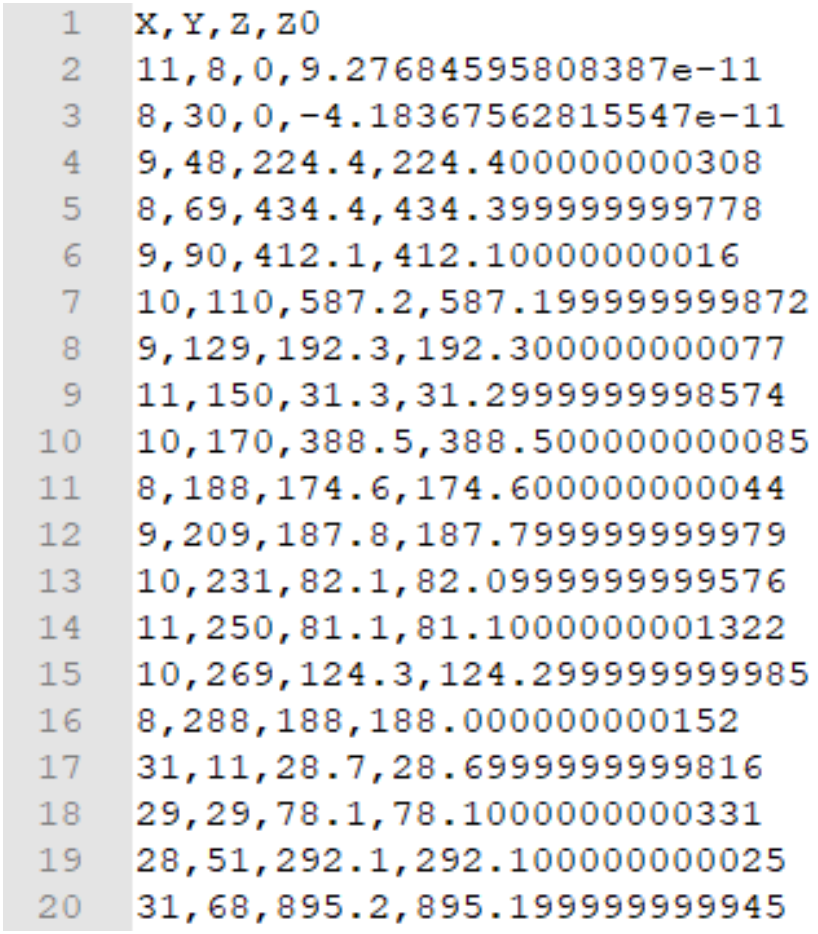


RESULTADO


Y0
0
50
100
150
200
250
300
X0
0
50
100
150
200
250
1287.694
1224.176
1160.657
1097.139
1033.620
970.102
906.583
843.065
779.546
716.027
652.509
588.990
525.472
461.953
398.435
334.916
271.398
207.879
144.361
80.842
17.323
-46.195

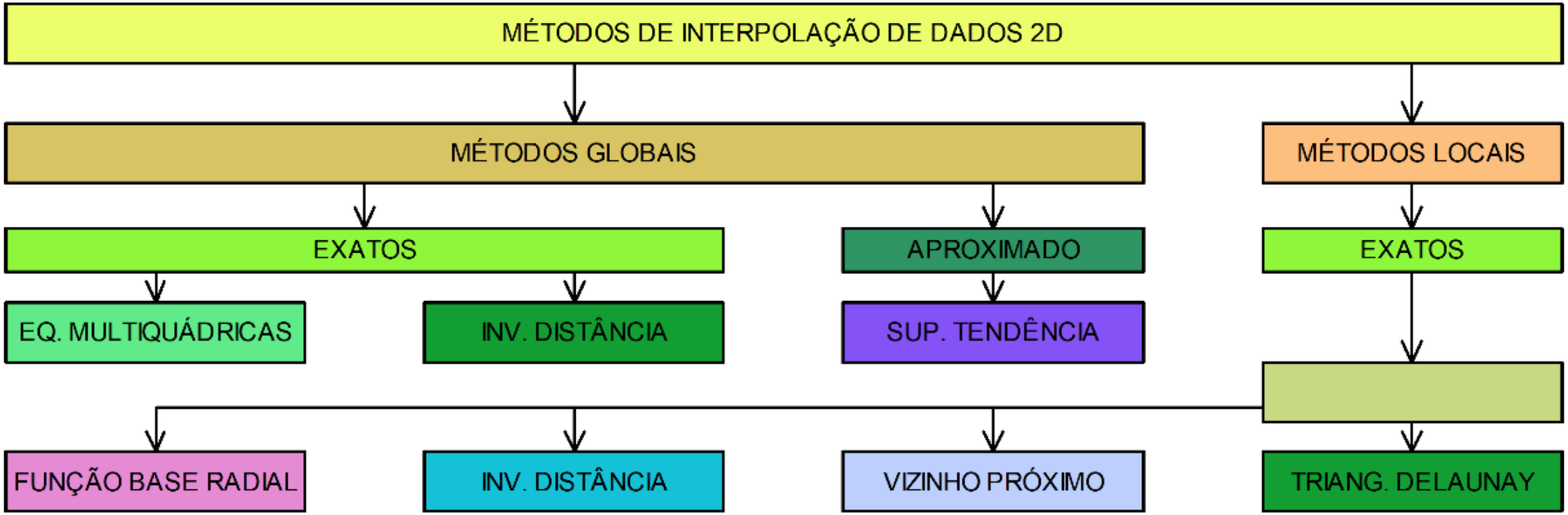
DISCUSSÃO

O resultado obtido mostra um ajuste respeitando as feições topográficas dos dados originais. Porém, deve-se comentar que o limite inferior negativo -46,195 era indesejável. Esse valor ocorre devido à resolução de um sistema de grandes dimensões 470 X 470, que pode ocasionar erros de arredondamento e truncamento. Esse efeito pode ser verificado no arquivo ao lado, onde calculamos os valores sobre os pontos de dados. Conclui-se que o ajuste pode ser considerado exato (passando pelos pontos), mas os pequenos resíduos se acumulam e se propagam. Entretanto, quando o número de pontos de dados for menor, o método das equações multiquádricas globais dá sempre um excelente resultado. Evidentemente, nesse caso se pode aplicar outros métodos exatos como, por exemplo, funções de base radial ou triangulação de Delaunay.



```
X,Y,Z,Z0
11,8,0,9.27684595808387e-11
8,30,0,-4.18367562815547e-11
9,48,224.4,224.400000000308
8,69,434.4,434.399999999778
9,90,412.1,412.10000000016
10,110,587.2,587.199999999872
9,129,192.3,192.300000000077
11,150,31.3,31.2999999998574
10,170,388.5,388.500000000085
8,188,174.6,174.600000000044
9,209,187.8,187.799999999979
10,231,82.1,82.0999999999576
11,250,81.1,81.1000000001322
10,269,124.3,124.299999999985
8,288,188,188.000000000152
31,11,28.7,28.6999999999816
29,29,78.1,78.1000000000331
28,51,292.1,292.100000000025
31,68,895.2,895.199999999945
```

OUTROS MÉTODOS DE INTERPOLAÇÃO

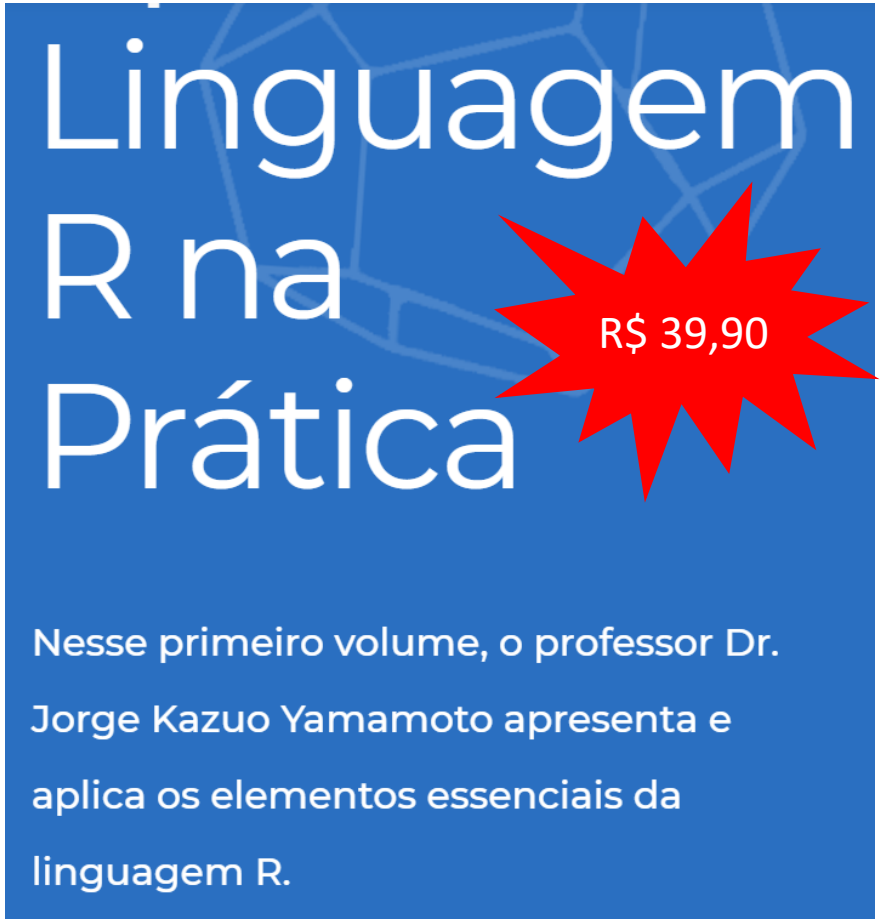


Todos estes métodos (globais ou locais) são acompanhados de scripts em R!
(Yamamoto, 2020, p. 228-300)

QUER APRENDER A LINGUAGEM R?

Temos um curso on-line e um livro para oferecer:

- https://geokrigagem.com.br/curso-online-linguagem-r-na-pratica/
- https://geokrigagem.com.br/produtos/estatistica-analise-e-interpolacao-de-dados-geoespaciais/



Preço normal:
R$ 189,90+21,00=210,90
Preço com desconto:
R$ 210,90-40,00=170,90
USE O CUPOM GKOFF40
VÁLIDO ATÉ O DIA 18/04/21

ISBN 978-65-990727-2-7
Editora: Gráfica Paulo's
8 capítulos em 308 páginas
1 capítulo bônus – Elementos de programação em linguagem R – 36 páginas
Data de publicação: setembro de 2020

REFERÊNCIAS BIBLIOGRÁFICAS

Benjamin, J.R.; Cornell, C.A. 1970. Probability, statistics and decision for civil engineers. Mineola, Dover Publications Inc. 684p.

Francis, A. 2004. Business mathematics and statistics. Hampshire, South-Western. 665p.

Franke, R. 1982. Scattered data interpolation: test of some methods. Math. of Computation. V. 38, p. 181-200.

Haan, C.T. 1977. Statistical methods in hydrology. Ames, The Iowa State University Press. 378p.

Hardy, R.L. 1971. Multiquadric equations of topography and other irregular surfaces. J. Geophysical Research, V. 76, p. 1905-1915.

Isaaks, E.H.; Srivastava, R.M. 1989, Applied geostatistics, New York, Oxford University Press. 561p.

Yamamoto, J.K. 2020. Estatística, análise e interpolação de dados geoespaciais. São Paulo, Gráfica Paulo's. 344p.

MUITO OBRIGADO!

Agradecimentos à Geocast Brasil,
em nome de seus organizadores:

- Felipe Sodré Mendes Barros
- Franklin J. Silva
- Kyle Felipe
- Luis Sadeck
- Maurício Humberto Vancine
- Narcélio de Sá