

# Atv12

Davi Ruas

## Análise Exploratória Gráfica Multivariada (final):

Lemos os dados do arquivo .txt:

```
library(readr)
library(gplots)
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##      lowess
```

```
geyser <- dget("~/Ufmg/2022-1/Pacotes/Atividades/08-07-22/geyser.txt")
geyser$waiting
```

```
##      [1] 80 71 57 80 75 77 60 86 77 56 81 50 89 54 90 73 60 83
##     [19] 65 82 84 54 85 58 79 57 88 68 76 78 74 85 75 65 76 58
##     [37] 91 50 87 48 93 54 86 53 78 52 83 60 87 49 80 60 92 43
##     [55] 89 60 84 69 74 71 108 50 77 57 80 61 82 48 81 73 62 79
##     [73] 54 80 73 81 62 81 71 79 81 74 59 81 66 87 53 80 50 87
##     [91] 51 82 58 81 49 92 50 88 62 93 56 89 51 79 58 82 52 88
##    [109] 52 78 69 75 77 53 80 55 87 53 85 61 93 54 76 80 81 59
##    [127] 86 78 71 77 76 94 75 50 83 82 72 77 75 65 79 72 78 77
##    [145] 79 75 78 64 80 49 88 54 85 51 96 50 80 78 81 72 75 78
##    [163] 87 69 55 83 49 82 57 84 57 84 73 78 57 79 57 90 62 87
##    [181] 78 52 98 48 78 79 65 84 50 83 60 80 50 88 50 84 74 76
##    [199] 65 89 49 88 51 78 85 65 75 77 69 92 68 87 61 81 55 93
##    [217] 53 84 70 73 93 50 87 77 74 72 82 74 80 49 91 53 86 49
##    [235] 79 89 87 76 59 80 89 45 93 72 71 54 79 74 65 78 57 87
##    [253] 72 84 47 84 57 87 68 86 75 73 53 82 93 77 54 96 48 89
##    [271] 63 84 76 62 83 50 85 78 78 81 78 76 74 81 66 84 48 93
##    [289] 47 87 51 78 54 87 52 85 58 88 79
```

```
geyser$duration
```

```
##      [1] 4.0166667 2.1500000 4.0000000 4.0000000 4.0000000 2.0000000 4.3833333
##      [8] 4.2833333 2.0333333 4.8333333 1.8333333 5.4500000 1.6166667 4.8666667
```

```
## [15] 4.3833333 1.7666667 4.6666667 2.0000000 4.7333333 4.2166667 1.9000000
## [22] 4.9666667 2.0000000 4.0000000 2.0000000 4.0000000 2.8333333 4.5000000
## [29] 4.0666667 3.7166667 3.5166667 4.4666667 2.2166667 4.8833333 2.6000000
## [36] 4.1500000 2.2000000 4.7666667 1.8333333 4.6000000 2.2666667 4.1333333
## [43] 2.0000000 4.0000000 2.0000000 4.0000000 1.8833333 4.2666667 2.0833333
## [50] 4.4666667 2.5000000 4.0000000 1.7666667 4.3333333 2.1833333 4.4833333
## [57] 3.8833333 3.3333333 3.7333333 4.0000000 1.9500000 5.2666667 2.0000000
## [64] 4.0000000 2.0000000 4.0000000 2.0000000 4.0000000 3.5333333 2.1666667
## [71] 4.5000000 2.0166667 4.1500000 4.2000000 4.3333333 1.9333333 4.6500000
## [78] 3.8166667 4.0333333 4.1666667 4.6666667 1.8166667 4.0000000 3.0000000
## [85] 4.0000000 2.0000000 4.4500000 2.0500000 4.2500000 1.9166667 4.6666667
## [92] 1.7333333 4.3833333 1.7666667 4.6000000 1.8666667 4.4500000 1.6333333
## [99] 5.0333333 1.8166667 5.1000000 1.6333333 4.2833333 2.0000000 4.0000000
## [106] 2.0000000 4.5333333 2.0000000 4.0000000 2.9333333 4.7333333 3.9000000
## [113] 1.9500000 4.1166667 1.8000000 4.6666667 1.8333333 4.7000000 2.1166667
## [120] 4.7833333 1.8166667 4.1000000 4.6500000 4.0000000 2.0000000 4.0000000
## [127] 4.0000000 4.2166667 4.1333333 3.9333333 3.7500000 4.4166667 2.4666667
## [134] 4.1666667 3.8000000 4.3166667 3.8666667 4.6833333 1.7000000 4.9666667
## [141] 4.2666667 4.5833333 4.0000000 4.0000000 4.0000000 4.0000000 1.9833333
## [148] 4.6000000 0.8333333 4.9166667 1.7333333 4.5833333 1.7000000 4.7500000
## [155] 1.8333333 4.5000000 1.8666667 4.4500000 4.4500000 4.0000000 4.8000000
## [162] 4.0000000 4.0000000 2.0000000 4.0000000 1.9333333 4.5833333 2.0000000
## [169] 3.7000000 2.8666667 4.8333333 3.4500000 4.3833333 1.8000000 4.4000000
## [176] 2.4833333 4.5166667 2.1000000 4.3500000 4.3666667 1.7833333 4.9166667
## [183] 1.8166667 4.0000000 4.0000000 4.0000000 3.8666667 1.8500000 4.7000000
## [190] 2.0166667 4.4666667 1.8666667 4.1666667 1.9000000 4.2500000 3.2500000
## [197] 4.2166667 1.8833333 4.9833333 1.8500000 4.0000000 1.9666667 4.7666667
## [204] 4.0000000 2.0000000 4.0000000 4.0000000 2.3833333 4.4166667 4.2166667
## [211] 4.3666667 2.0000000 4.4500000 1.7500000 4.5000000 1.6166667 4.7000000
## [218] 2.5666667 3.7000000 4.2333333 1.9333333 4.3500000 4.0000000 4.0000000
## [225] 4.0000000 4.2166667 4.0000000 4.1333333 1.8833333 4.4666667 1.9500000
## [232] 4.2166667 1.7166667 4.4500000 4.2500000 3.9666667 4.3833333 1.9666667
## [239] 4.4500000 4.2666667 1.9166667 4.4166667 3.0000000 4.0000000 2.0000000
## [246] 4.0000000 3.2833333 1.8333333 4.6166667 1.8333333 4.6166667 4.6000000
## [253] 4.2500000 1.9333333 4.9833333 1.9666667 4.3000000 4.2000000 4.5333333
## [260] 4.4000000 4.6166667 2.0000000 4.0000000 4.0000000 3.9166667 2.0000000
## [267] 4.5000000 1.8000000 4.0000000 2.7500000 4.7333333 3.9666667 1.9500000
## [274] 4.9666667 1.8500000 4.8000000 4.0000000 4.0000000 4.0000000 4.0000000
## [281] 4.0000000 4.0000000 4.0000000 2.0000000 4.0000000 1.9333333 4.3333333
## [288] 1.6666667 4.7666667 1.9500000 4.6833333 1.9333333 4.4166667 2.1333333
## [295] 4.0833333 2.0666667 4.0000000 4.0000000 2.0000000
```

```
attach(geyser)
```

Fazendo as análises descritivas e o grafico das variaveis:

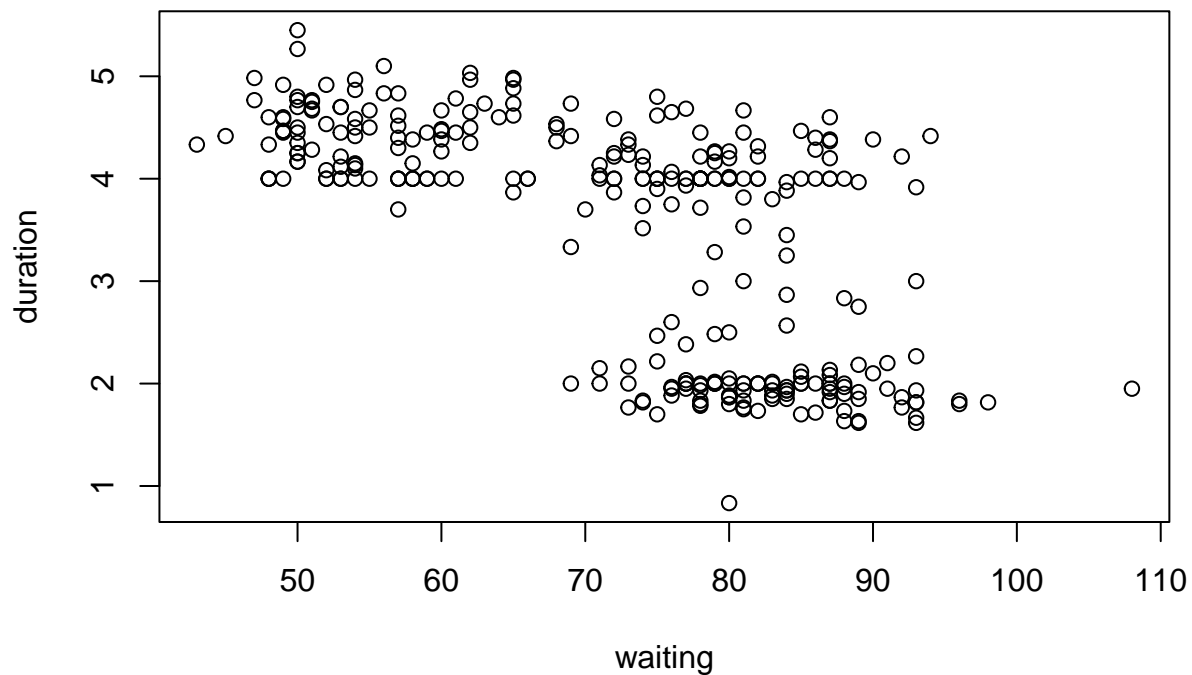
```
summary(waiting)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    43.00   59.00   76.00   72.31   83.00   108.00
```

```
summary(duration)
```

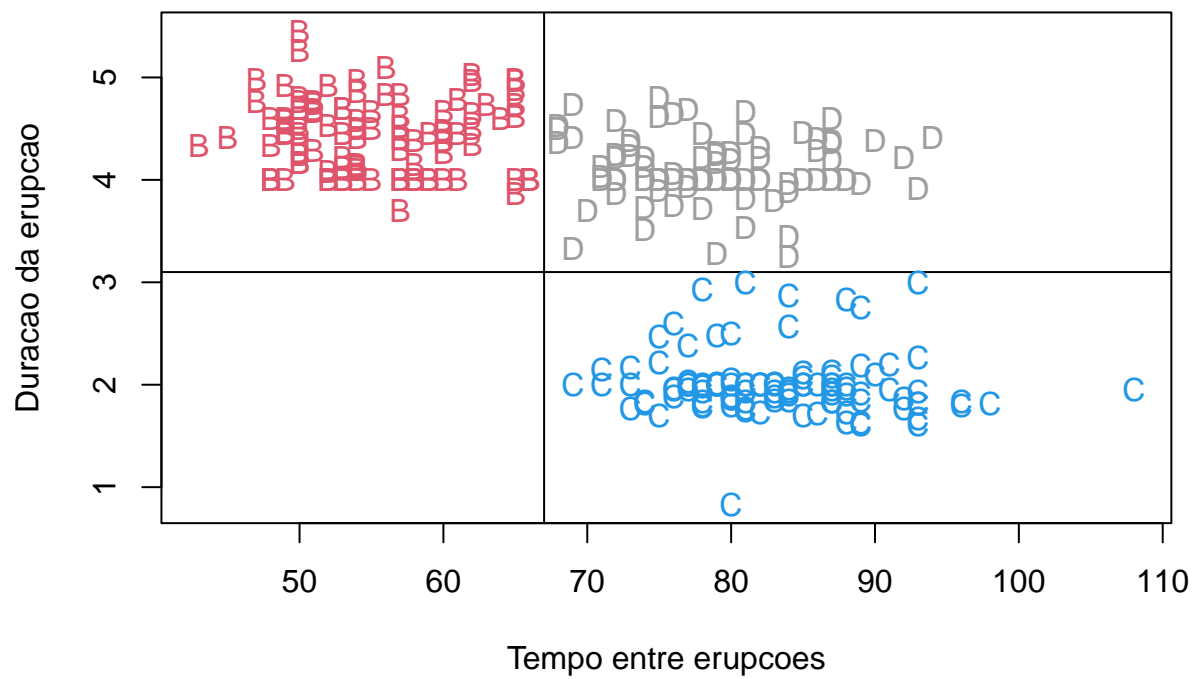
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.8333  2.0000  4.0000  3.4608  4.3833  5.4500
```

```
plot(waiting, duration)
```



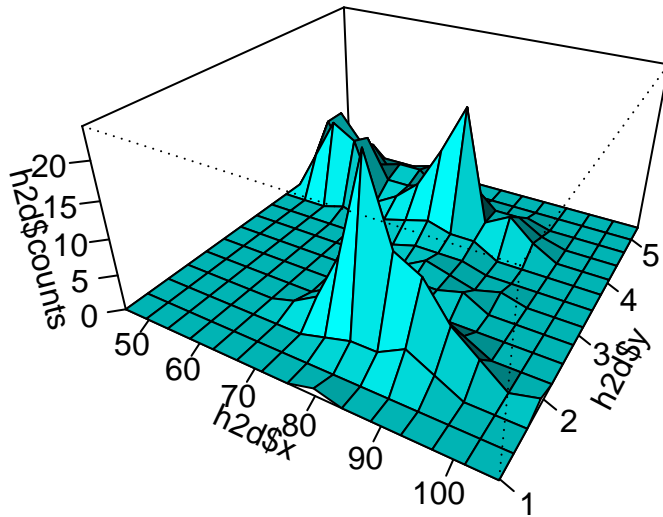
Gerando graficos com clusters:

```
plot(waiting, duration, type="n", ylab="Duracao da erupcao", xlab="Tempo entre erupcoes")
abline(h=3.1)
abline(v=67)
subset.1 <- (waiting < 67) & (duration < 3.1)
subset.2 <- (waiting < 67) & (duration > 3.1)
subset.3 <- (waiting > 67) & (duration < 3.1)
subset.4 <- (waiting > 67) & (duration > 3.1)
points(waiting[subset.1], duration[subset.1], pch="A", col=1)
points(waiting[subset.2], duration[subset.2], pch="B", col=2)
points(waiting[subset.3], duration[subset.3], pch="C", col=4)
points(waiting[subset.4], duration[subset.4], pch="D", col=8)
```



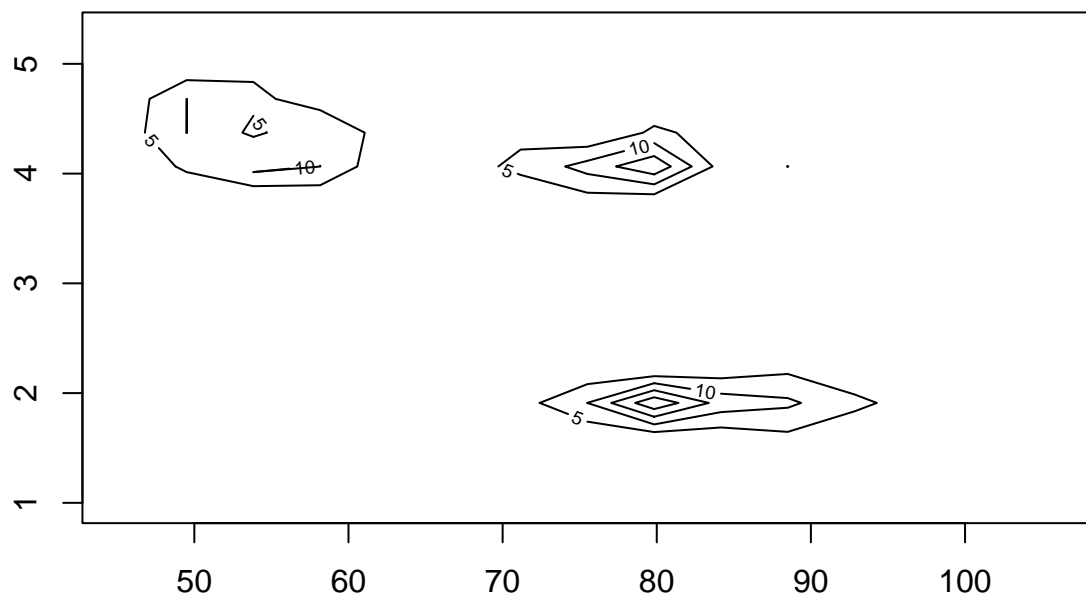
Gerando gráfico 3D:

```
h2d <- hist2d(waiting, duration, show=FALSE, same.scale=FALSE, nbins=c(15,15))
persp(h2d$x, h2d$y, h2d$counts, ticktype="detailed", theta=30, phi=30, expand=0.5, shade=0.5, col="cyan")
```

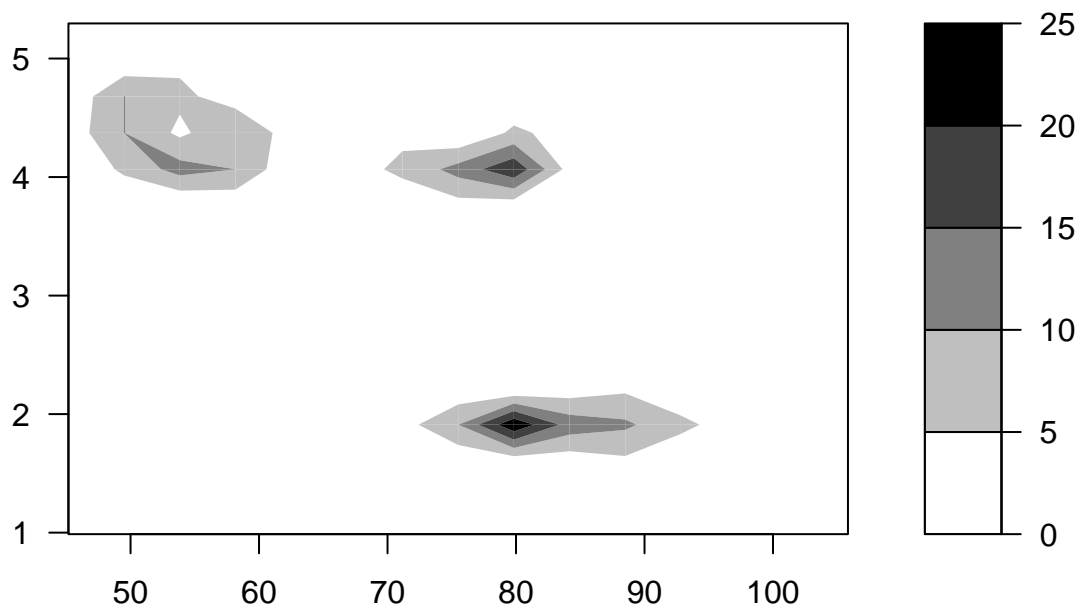


Gerando gráfico de curvas de nível, com e sem cores(tons de cinza):

```
contour(h2d$x, h2d$y, h2d$counts, nlevels=4)
```



```
filled.contour(h2d$x, h2d$y, h2d$counts, nlevels=4, col=gray((4:0)/4) )
```



## Analizando dados ausentes:

Tente:

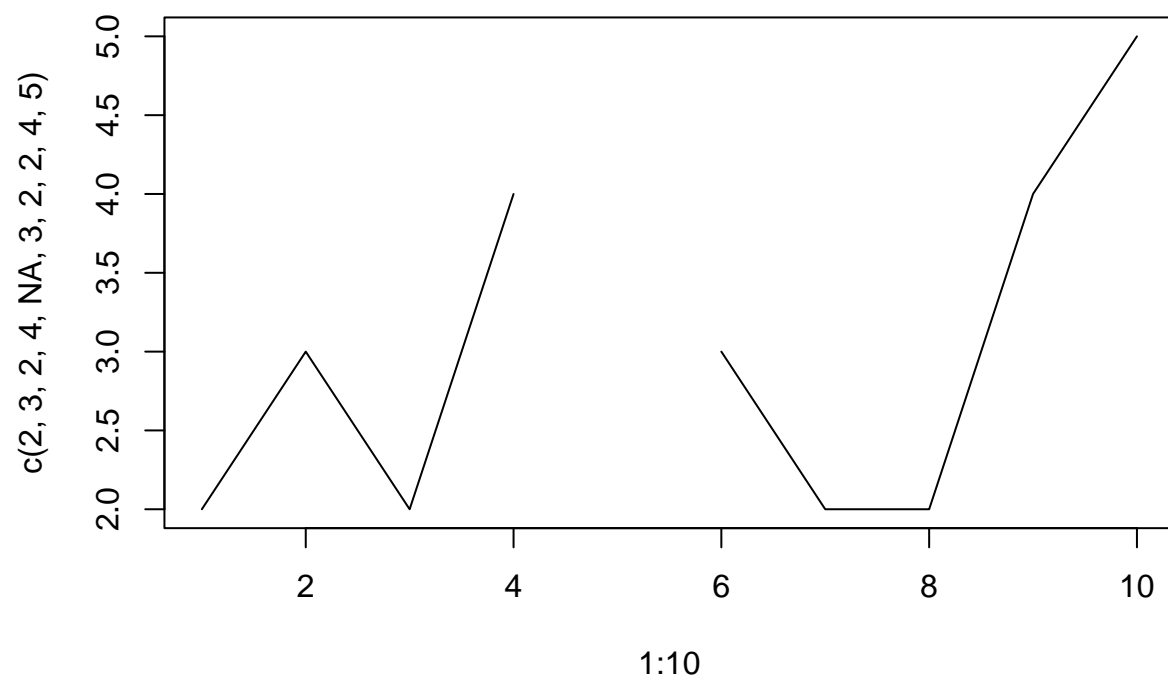
```
x <- c(1, 2, NA, 4)

cat(c('\nMedia com NA:', mean(x), '\n',
      '\nMedia com a função "na.rm" que remove os dados NA', mean(x, na.rm=T), '\n',
      '\nCalcula a media baseado em vetor onde não há dados NA', mean(x[!is.na(x)]), '\n'))
```

```
##
## Media com NA: NA
##
## Media com a função "na.rm" que remove os dados NA 2.33333333333333
##
## Calcula a media baseado em vetor onde não há dados NA 2.33333333333333
```

Tente também:

```
plot(1:10, c(2,3,2,4,NA,3,2,2,4,5), type="l")
```



Finalmente, teste:

```
x <- (-1:1)/0
x
```

```
## [1] -Inf NaN Inf
```

```
is.na(x)
```

```
## [1] FALSE TRUE FALSE
```

```
is.infinite(x)
```

```
## [1] TRUE FALSE TRUE
```