Universidade Federal do Rio de Janeiro
Observatório do Valongo

Segunda prova

Astroestatística

2025.1

Iago Lopes
DRE: 122077032

Professor: Hélio Jaques

17 de Julho 2025

Questão 1 - Represente cada aluno por uma Face de Chernoff. A partir delas, você julga que há alunos que apresentem um histórico no ENEM e CRA do primeiro período similar entre si? Ou não?

Um bom método para exploração e classificação é o de Faces de Chernoff que utiliza característica físicas para exemplificar o quão similar os dados são no espaço de seus parâmetros. Duas faces similares indicam pontos que estão em uma região próxima no espaço dos parâmetros. Esse método facilita para humanos avaliarem similaridades nos dados e então poder trabalhar em cima dessa análise inicial. Para isso usei a função `faces` considerando todas as colunas da amostra de alunos da astronomia, exceto seus nomes e o ano da turma, já que não são relevantes ao olhar para a nota do Enem e o CRA.

```
install.packages('TeachingDemos')
library(TeachingDemos)

astro = read.table('/content/ENEM_Astro.csv', sep=',', header=T)

options(repr.plot.width=14,repr.plot.height=12)
# Usando todas colunas, exceto o nome para criar as faces
faces(astro[, 2:7], labels = astro$Nome)
```

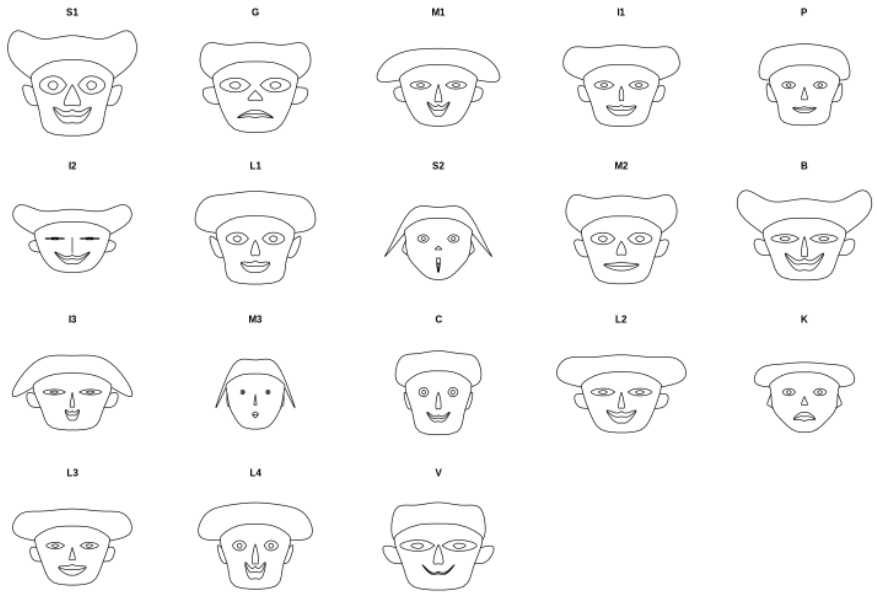

Figura 1: Resultado ao aplicar o método de faces de Chernoff na amostra de alunos da astronomia.

O resultado pode ser visto na imagem 1, onde é possível dizer que de fato há alunos similares nesse espaço de parâmetros. Os principais grupo que percebi foi o de: (S1, B), (S2, M3) e (L1, L2, L3) como indicado na imagem.

Questão 2 - Com base nessas notas, você diria que o CRA do primeiro período é uma função linearmente proporcional ao desempenho do aluno no ENEM?

Para analisar rigorosamente, irei fazer um ajuste linear simples e obter o valor p. A minha métrica de desempenho para o ENEM será apenas a média ponderada, onde assumirei peso igual para todas matérias.

```
install.packages('MASS')
library(MASS)

# Data
enem_mean = rowMeans(astro[,c(2:6)])
cra = astro$CRA

# Reg
reg = lm(cra ~ enem_mean,method='MM')
summary(reg)

### output ###
Call:
lm(formula = cra ~ enem_mean, method = "MM")

Residuals:
    Min      1Q  Median      3Q     Max
-3.0514 -0.9285  0.1149  1.2654  1.8299

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.965886   5.982551  -1.332   0.2017
enem_mean    0.021012   0.008508   2.470   0.0252 *
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1
       1

Residual standard error: 1.525 on 16 degrees of freedom
Multiple R-squared:  0.276, Adjusted R-squared:  0.2308
F-statistic:   6.1 on 1 and 16 DF,  p-value: 0.02515

# Plot
x = c(min(enem_mean):max(enem_mean),100)
plot(enem_mean, cra, cex=2.5, col='red',pch=16,xlab='M dia ENEM',
     ylab='CRA',cex.lab=2.5,cex.axis=2)
abline(reg, col='blue', lwd=3)
legend('topleft',legend=c('Alunos','Ajuste linear'),col=c('red','blue'),
pch=c(16,NA),lty = c(NA, 1),lwd = c(NA, 3),cex=c(1.5,1.5),bty='n')

# Text
text(640, 3.5, paste('Correlacao (Spearman):', correlation, "\n p-valor:",
 round(test$p.value,3)),
     cex=2.5,
     font=2)
```

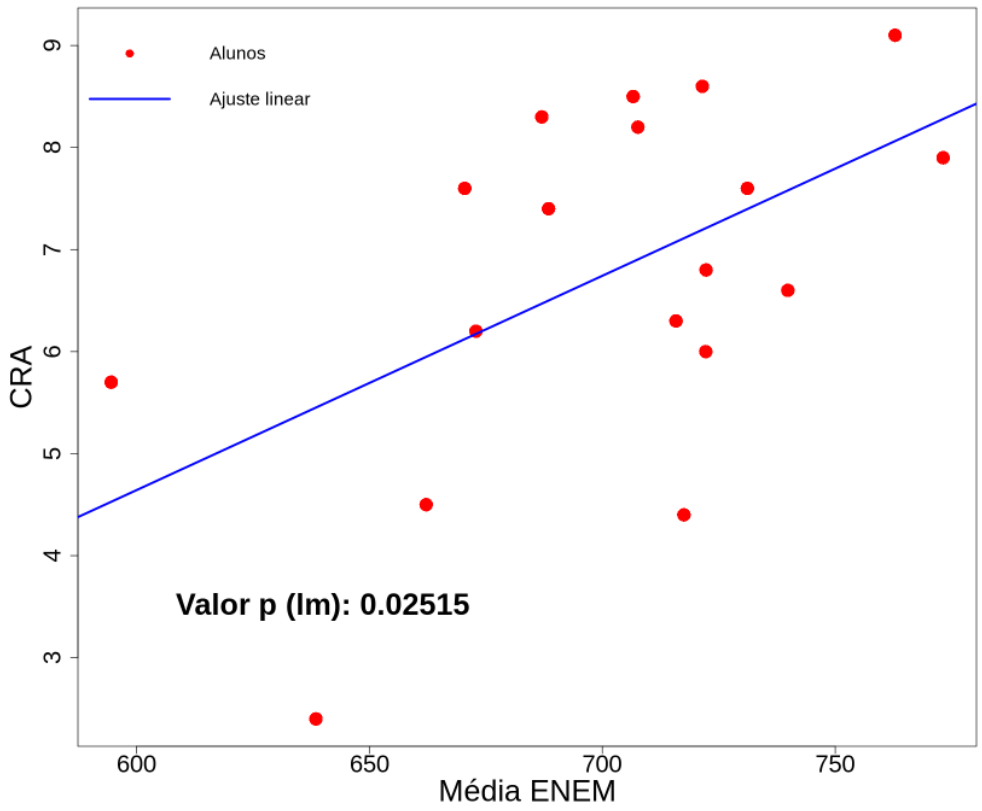

Figura 2: CRA em função da média ponderada no ENEM dos alunos da astronomia das turmas de 2024 e 2025.

Ao observar o valor p do modelo linear na figura 2, não podemos fazer nenhuma afirmação sobre a linearidade entre o CRA e o desempenho no ENEM.

Questão 3 - Represente a separação entre esses alunos em um plano bidimensional.

Uma maneira de representar a separação dos alunos em relação ao espaço dos parâmetros em dimensões menores é computar a matriz de distâncias no espaço dos parâmetros e então usá-la para projetar em dimensões menores. Primeiro, normalizei as quantidades de diferentes colunas para uma melhor comparação entre o CRA, média no ENEM e o ano de ingresso. Então, calculei a matriz de distância e projetei em 2 dimensões.

```
# Distance matrix
m_dist = dist(scale(astro[,c(2:8)]))

# Projecting on 2 dim
projected = cmdscale(m_dist)

# Plots
plot(projected[,1], projected[,2], cex=2,col='red',
pch=16,xlab='Projecao 1', ylab="Projecao 2", cex.lab=2)
text(projected[,1], projected[,2],
     labels = astro$Nome,
     pos = 3,
     cex = 1.5)
```

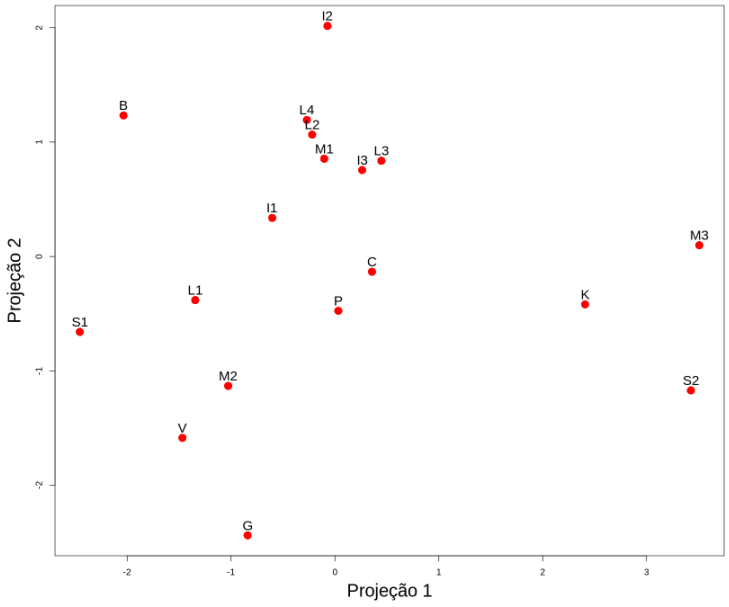

Figura 3: Projeção do espaço de parâmetro dos alunos em 2 dimensões com o respectivo nome acima.

Questão 4 - Compare a densidade das notas obtidas pelos alunos de 2024 com as de 2025 em uma figura com 6 paineis, um para cada nota. As distribuições são estatisticamente similares entre os alunos de cada ano?.

Se buscamos verificar a similaridade estatística entre as distribuições de cada ano, precisamos de testes estatísticos que nos permitam realizar afirmações. Nessa sitaução, utilizei 4 testes: t-test, F-test, KS-test, Wilcox-test. A escolha dos testes foi feita de maneira que seja possível testes diferentes características das amostras. Além disso, decidir fazer um teste utilizando a MANOVA, ou seja, considerando a mediana de todas as dimensões ao mesmo tempo e obtive um valor p de: 0.49

```
par(mfcol = c(3, 2))

colors = c('blue', 'red')
for (i in 2:7) {
  values_1 = astro[astro$ano == 2025, i]
  values_2 = astro[astro$ano == 2024, i]

  # Tests
  t_test = t.test(values_1, values_2)
  f_test = var.test(values_1, values_2)
  ks_test = ks.test(values_1, values_2)
  wil = wilcox.test(values_1, values_2, exact = FALSE)

  # Plot
  plot(density(values_1), col = colors[1], xlab = 'Pontua  o',
       main = colnames(astro)[i], ylim = c(0, max(density(values_1)$y,
        density(values_2)$y) * 1.2))
  lines(density(values_2), col = colors[2])

  # Add text
  text_x = median(median(values_1),median(values_2)) *0.8
  y_start = 0.008

  text(text_x, y_start*0.9,     paste("t: ", round(t_test$p.value, 3),
  "\nF: ", round(f_test$p.value, 3),
  "\nKS: ", round(ks_test$p.value, 3),
  "\nWil: ", round(wil$p.value, 3)), pos = 3)
}

legend('topleft', legend = c(2025, 2024), col = colors, lty = 1, bty = 'n',
cex = 1.2)
# MANOVA
man_test = manova(as.matrix(astro[,2:7]) ~ astro$ano)
cat("\nMANOVA teste para todas as  reas :\n")
print(summary(man_test))
```

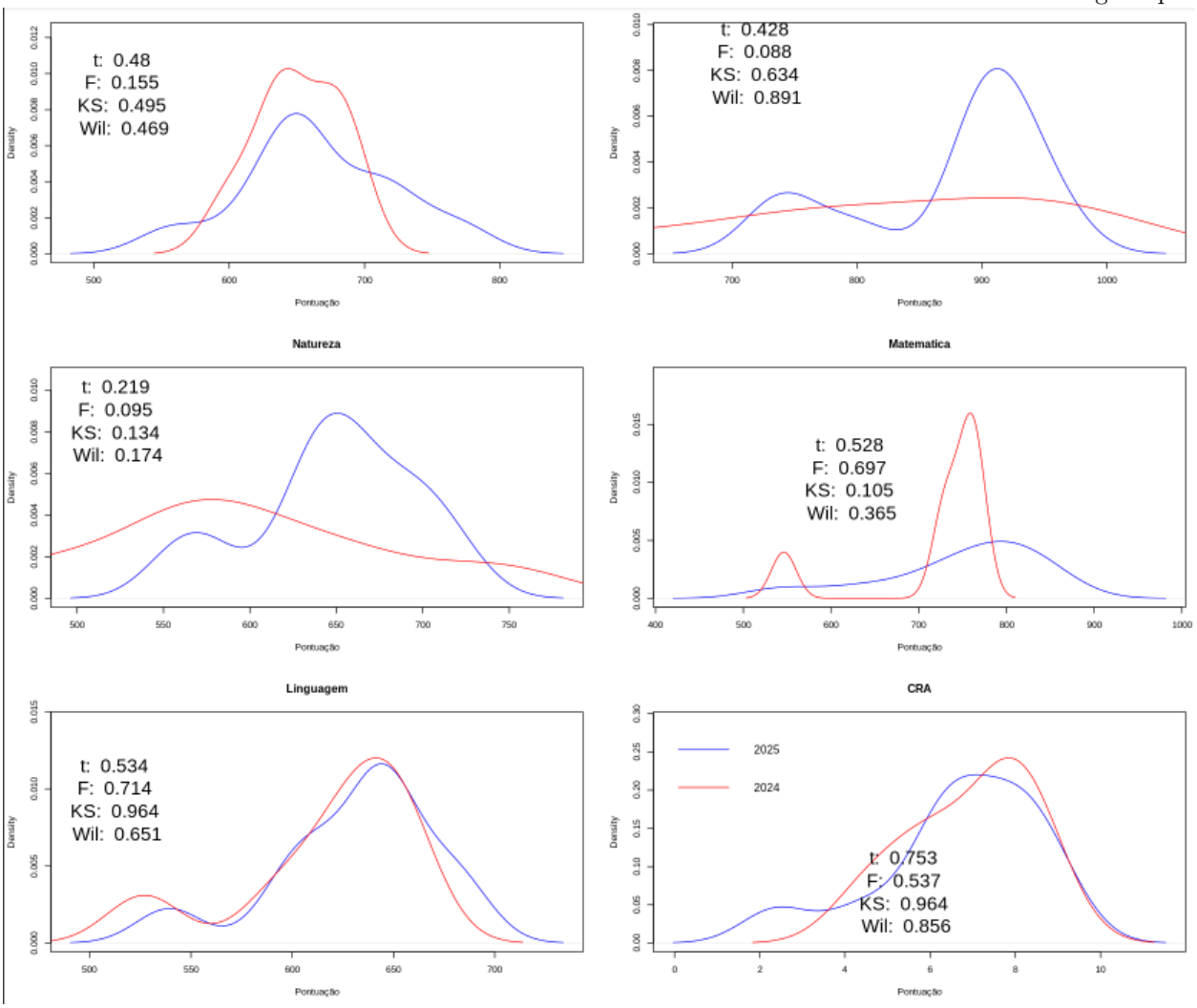

Figura 4: Distribuição da notas dos alunos da astronomia classificados por ano de ingresso junto com o valor p dos 4 testes estatísticos escolhidos para a análise.

Ao olhar o resultado de diferentes testes na figura 4, não podemos afirmar que as distribuições são de diferentes populações. Portanto, pode-se dizer que são estatisticamente similares.

Questão 5 - A galáxia NGC 5548 hospeda um AGN classificado como Seyfert 1. Peterson et al. (1999) observaram seu fluxo durante 8 anos no contínuo óptico (em 5100 Å) e na linha Hβ. O arquivo `NGC5548.dat` contém a data juliana da observação (JD), os fluxos no contínuo óptico e na linha Hβ (F5100 e FHbeta), bem como seus respectivos erros (e_F5100 e e_FHbeta). Os fluxos estão medidos em erg cm^{-2} s^{-1} Å^{-1}.

1. Suponha que seja possível descrever o fluxo no contínuo óptico deste AGN a partir do fluxo na linha Hβ, e que apenas essas duas informações sejam conhecidas. Que parametrização seria essa?
2. Suponha agora que as medidas de erro sejam conhecidas. Critique o resultado obtido acima com base no que esses erros indicam. Represente ambas as distribuições empíricas em um gráfico quantil-quantil.
3. Conhecendo os valores medidos para os fluxos e seus erros estimados, sugira uma parametrização mais adequada. Explique por que ela é mais adequada a esse problema e como ela foi obtida.

A) Essa parametrização seria uma regressão linear ordinária, onde não leva-se em conta possíveis erros nas variáveis dependente e independente. Para isso, usei o linear model (lm) do R.

```
ngc = read.table('/content/NGC5548.dat', sep='|', header = T)
ngc = ngc[complete.cases(ngc),]

options(repr.plot.width=10,repr.plot.height=10)

fit = lm(F5100 ~ FHbeta, data = ngc)

x_vals = seq(min(ngc$FHbeta), max(ngc$FHbeta), length.out = 200)
y_pred = predict(fit, newdata = data.frame(FHbeta = x_vals))

plot(ngc$FHbeta, ngc$F5100, pch=20, xlab="FHbeta", ylab="F5100",
xlim=c(6,14), ylim=c(7,15))
lines(x_vals, y_pred, col = "blue", lwd = 2)

for (i in 1:length(ngc$FHbeta)) {
  lines(
    c(ngc$FHbeta[i], ngc$FHbeta[i]),
    c(ngc$F5100[i] + ngc$e_F5100[i], ngc$F5100[i] - ngc$e_F5100[i]),
    col=gray(0.5)
    )
  lines(
    c(ngc$FHbeta[i] + ngc$e_FHbeta[i], ngc$FHbeta[i] - ngc$e_FHbeta[i]),
    c(ngc$F5100[i], ngc$F5100[i]),
    col=gray(0.5)
    )
}
```

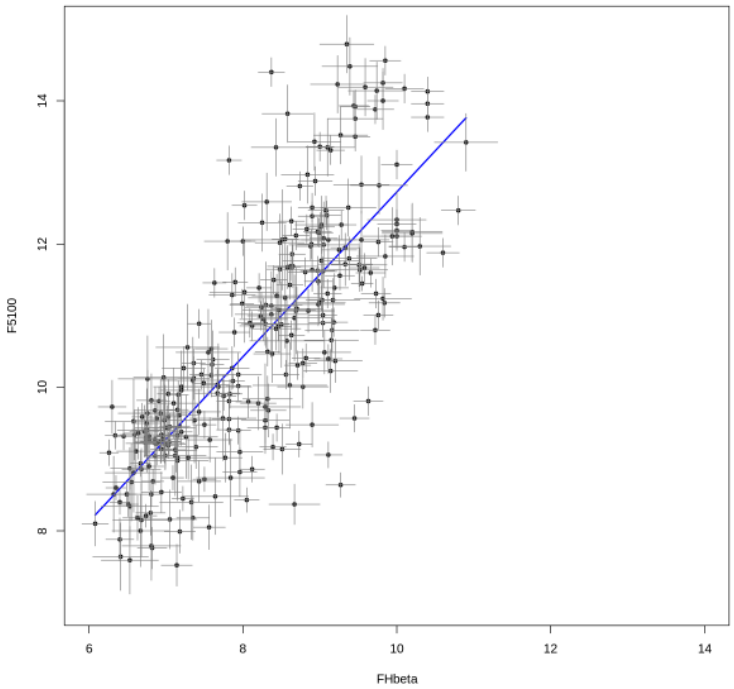

Figura 5: Regressão linear ordinária do fluxo contínuo no ótico em relação ao fluxo na linha Hβ em azul. Os pontos mostram os dados com as barras de erro.

B) Se as medidas de erro são conhecidas, então devemos utilizar um método que leve em conta possíveis erro nas medidas. Além disso, deve-se usar a variável com menor erro como variável independente e para verificar qual tem menor erro, eu calculei o RMS do erro relativo de ambas quantidades. Logo, deve-se utilizar o fluxo em contínuo no ótico como variável independente.

```
rms_rel_F5100 = sqrt(mean((ngc$e_F5100 / ngc$F5100)^2))
rms_rel_FHbeta = sqrt(mean((ngc$e_FHbeta / ngc$FHbeta)^2))

cat("Erro RMS relativo em F5100:", round(rms_rel_F5100, 3), "\n")
cat("Erro RMS relativo em FHbeta:", round(rms_rel_FHbeta, 3), "\n")

### output ###
Erro RMS relativo em F5100: 0.033
Erro RMS relativo em FHbeta: 0.034
```

C) A parametrização mais adequada para esse caso é o método de mínimos quadrados pesado pelo erro e utilizando o valor com menor erro como variável independente. Para isso, usei o linear model (lm) usando os erros como peso.

```
weight = 1 / (ngc$e_FHbeta * ngc$e_FHbeta)

fit = lm(FHbeta ~ F5100, data = ngc, weights = weight)

x_vals = seq(min(ngc$F5100), max(ngc$F5100), length.out = 200)
y_pred = predict(fit, newdata = data.frame(F5100 = x_vals))

plot(ngc$F5100, ngc$FHbeta, pch = 20, ylab = "FHbeta", xlab = "F5100",
```

```
    ylim=c(6,14), xlim=c(7,15))
    lines(x_vals, y_pred, col = "blue", lwd = 2)

    for (i in 1:nrow(ngc)) {
      lines(
        c(ngc$F5100[i] - ngc$e_F5100[i], ngc$F5100[i] + ngc$e_F5100[i]),
        c(ngc$FHbeta[i], ngc$FHbeta[i]),
        col = gray(0.5)
      )

      lines(
        c(ngc$F5100[i], ngc$F5100[i]),
        c(ngc$FHbeta[i] - ngc$e_FHbeta[i], ngc$FHbeta[i] + ngc$e_FHbeta[i]),
        col = gray(0.5)
      )
    }
```

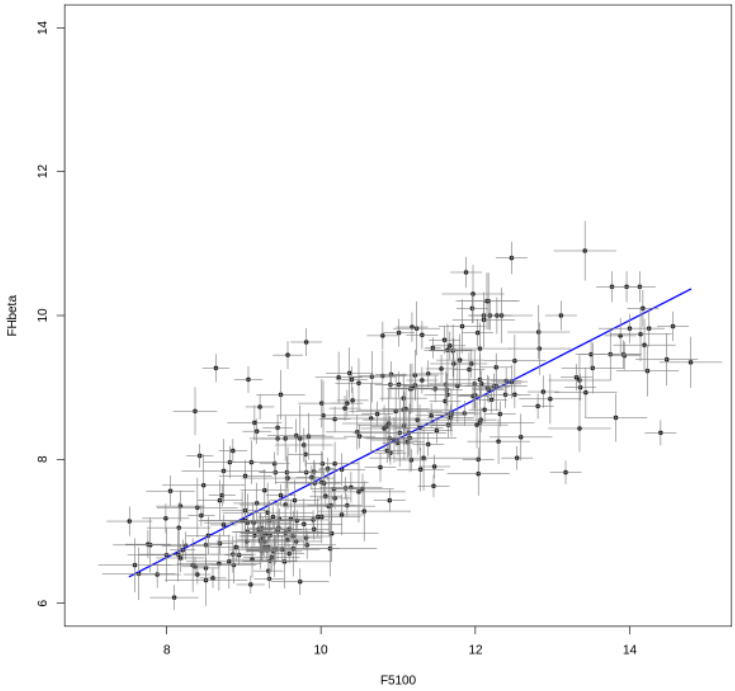

Figura 6: Regressão linear pesada pelo erro do fluxo na linha Hβ em relação ao fluxo contínuo no ótico em azul. Os pontos mostram os dados com as barras de erro.

Questão 6 - Transforme as cores em magnitudes U, B, R e I, e junte essas magnitudes à V em uma nova dataframe. Aplique a técnica de componentes principais nesta dataframe formada apenas por magnitudes aparentes.

1. Quantas componentes principais poderiam explicar apropriadamente a distribuição das estrelas nesse espaço de magnitudes?
2. Interprete. O que deve significar fisicamente os dois primeiros componentes principais?
3. Quais seriam as magnitudes de uma estrela hipotética que pudesse ser representada no espaço de componentes principais pelas coordenadas (0.2, 0.43, 0.6, 0.8, 0.4)?

O método de PCA rotaciona o espaço dos parâmetros de maneira que os primeiros eixos sejam aqueles onde há maior variância dos dados. Esse método é muito útil para reduzir a dimensionalidade do problema ao assumir que as componentes de menor variância são ruídos.

```
catalog = read.table('/content/king5.tsv', sep='|', header=T)

# Estimating magnitude
Bmag = catalog$BV+catalog$Vmag
Umag = catalog$UB+Bmag
Rmag = -(catalog$VR-catalog$Vmag)
Imag = -(catalog$VI-catalog$Vmag)

new_df = data.frame(Umag = Umag, Bmag = Bmag, Vmag = catalog$Vmag, Rmag = Rmag
    ,
 Imag = Imag
)
new_df = new_df[complete.cases(new_df),]

# PCA
pca = prcomp(new_df)
summary(pca)

### output ###
Importance of components:
                          PC1     PC2     PC3     PC4     PC5
Standard deviation     2.7876 0.42396 0.17145 0.04079 0.03337
Proportion of Variance 0.9735 0.02252 0.00368 0.00021 0.00014
Cumulative Proportion  0.9735 0.99597 0.99965 0.99986 1.00000
```

a) Vemos que as duas primeira componentes já conseguem explicar mais de 99,5 % da variância, portanto eu diria que duas componentes principais já são suficientes para explicar os dados.

```
par(mfrow=c(2,1))
barplot(pca$rotation[,1]) # Brilho
barplot(pca$rotation[,2]) # Cor
```

b) Para interpretar, decidi fazer um gráfico de barras das componentes e percebi que a primeira componente está relacionada simplesmente com a magnitude das estrelas, ou seja, o brilho aparente delas. Já para a segunda componente, notei que as banda U e I estão com maior peso, logo imagino que essa componente está buscando explicar as cores das estrelas, ou seja, suas temperaturas efetivas, pois se a T_{eff} é baixa, então emitirá mais em comprimentos de onda maiores, enquanto que para T_{eff} alta, temos maior emissão em comprimentos de onda menores.

c) Para computar as magnitudes nesse espaço, basta utilizar os coeficientes, ou seja, aplicar a transposta da matriz no vetor do espaço do PCA.

```
pca_vector = c(0.2, 0.43, 0.6, 0.8, 0.4)
mags = c()
for (i in c(1:5)){
  mag = sum(pca$rotation[i,] * pca_vector) + mean(new_df[,i])
  mags = c(mags,mag)
}
cat('Magnitudes (U,B,V,R,I): ',mags)

### output ###
Magnitudes (U,B,V,R,I):  19.74012 17.93687 17.82269 16.01448 16.04089
```

Questão 7 - Aplique uma decomposição de misturas às magnitudes U, B, V, R e I do aglomerado King 5 considerando que cada componente seja uma normal multivariacional.

1. Quantas componentes normais multivariacionais são necessárias para explicar a distribuição dessas magnitudes?
2. Quais são as magnitudes médias e a matriz de covariância de cada uma dessas componentes?
3. Faça dois diagramas cor—magnitude, um ao lado do outro. O primeiro deve ser BV × V; o segundo deve ser RI × R. Use o argumento col do comando plot para identificar cada objeto com base na classificação atribuída a ele pela decomposição de misturas.

a) De acordo com o modelo de decomposição de misturas, temos 3 componentes necessárias para explicar a distribuição.

```
install.packages('mclust')
library(mclust)
model = Mclust(new_df,modelNames = 'VVV')
summary(model)

### output ###
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 3
components:

 log-likelihood   n df       BIC      ICL
       618.4788 189 62 911.9692 886.6007

Clustering table:
 1  2  3
37 85 67
```

b) O vetor das médias pode ser visto no output do código, junto com as matrizes de covariância. Esses valores descrevem completamente as 3 gaussianas multivariacionais encontradas pelo mclust.

```
for (i in c(1:3)){
  cat('\nM dia da componente',i,':',model$parameters$mean[,i])
}
for (i in 1:3) {
  cat('Matriz componente', i, ':\n')
  print(model$parameters$variance$sigma[,,i])
  cat('\n')
}
```

```
### output ###
M dia da componente 1 : 19.08092 18.2044 16.9453 16.21884 15.41783
M dia da componente 2 : 18.29859 17.73985 16.61541 15.98515 15.26787
M dia da componente 3 : 20.14439 19.56443 18.22695 17.45339 16.62603

Matriz componente 1 :
         Umag     Bmag     Vmag     Rmag     Imag
Umag 3.645411 3.224176 2.959774 2.766449 2.650813
Bmag 3.224176 3.386730 3.149322 2.955543 2.866553
Vmag 2.959774 3.149322 2.993255 2.820833 2.780000
Rmag 2.766449 2.955543 2.820833 2.674180 2.646377
Imag 2.650813 2.866553 2.780000 2.646377 2.660136

Matriz componente 2 :
          Umag      Bmag      Vmag      Rmag      Imag
Umag 0.7108262 0.7305238 0.6625979 0.6187826 0.5740213
Bmag 0.7305238 0.7592262 0.6867012 0.6400591 0.5929057
Vmag 0.6625979 0.6867012 0.6250097 0.5855402 0.5447514
Rmag 0.6187826 0.6400591 0.5855402 0.5511798 0.5147319
Imag 0.5740213 0.5929057 0.5447514 0.5147319 0.4824006

Matriz componente 3 :
          Umag      Bmag      Vmag      Rmag      Imag
Umag 0.3867627 0.4024006 0.3650820 0.3445183 0.3291485
Bmag 0.4024006 0.4465300 0.4134285 0.3957304 0.3819135
Vmag 0.3650820 0.4134285 0.3919677 0.3794756 0.3695648
Rmag 0.3445183 0.3957304 0.3794756 0.3711844 0.3638454
Imag 0.3291485 0.3819135 0.3695648 0.3638454 0.3586400
```

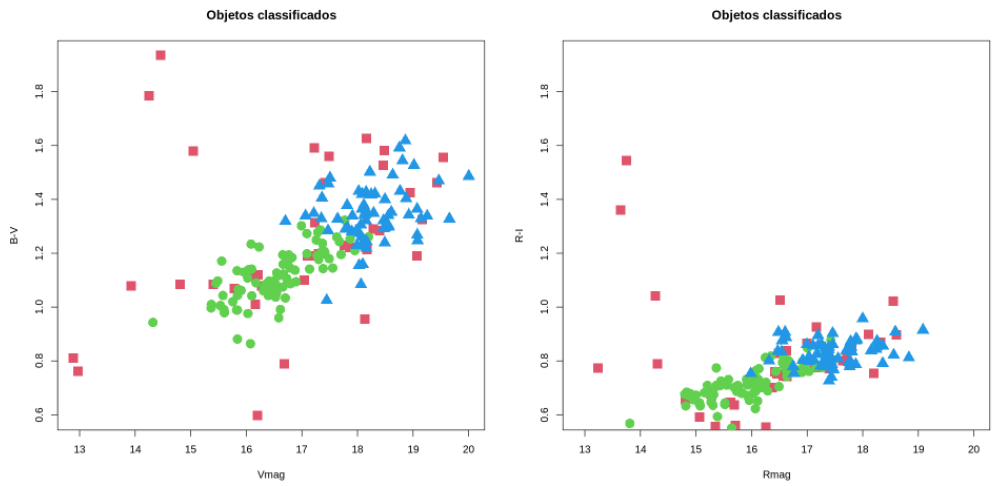

Figura 7: Estrelas do aglomerado aberto King 5 em dois diagramas cor x magnitude. As cores mostram o grupo classificado pela separação de misturas usando o mclust.

c) Podemos ver que de fato há 2 populações na imagem 7, onde os objetos do grupo em azul são estrelas mais avermelhadas e com menor brilho aparente, enquanto que o grupo verde são estrelas mais azuladas e maior brilho aparente. Como é um aglomerado e as estrelas devem estar aproximadamente a mesma distância,

podemos dizer que esse gráfico se aproxima bastante de um diagrama HR do aglomerado. O terceiro grupo parece ser construídos de outliers, possivelmente objetos fora do aglomerado ou estrelas fora da sequência principal.

```
options(repr.plot.width=16,repr.plot.height=8)
par(mfrow=c(1,2))
plot(c(), xlim=range(new_df$Vmag), ylim=range(new_df$Bmag - new_df$Vmag),
     xlab="Vmag", ylab="B-V", main="Objetos classificados")
for (i in c(1:3)){
  sub_df = new_df[model$classification==i,]
  points(sub_df$Vmag,sub_df$Bmag-sub_df$Vmag, col=1+i, pch=14+i,cex=2)
}

plot(c(), xlim=range(new_df$Vmag), ylim=range(new_df$Bmag - new_df$Vmag),
     xlab="Rmag", ylab="R-I", main="Objetos classificados")
for (i in c(1:3)){
  sub_df = new_df[model$classification==i,]
  points(sub_df$Rmag,sub_df$Rmag-sub_df$Imag, col=1+i, pch=14+i,cex=2)
}
```

Questão 8 - Tasse et al. (2011) mediu a emissão de raios x em AGNs para estudar a relação entre a atividade do AGN e a taxa de formação estelar na galáxia hospedeira. Seus dados se encontram no arquivo xrays_tasse.tsv. As colunas são nome do AGN (Name), log fluxo em raios X moles (logFxs), log fluxo em raios X duros (logFxh), magnitudes gmag, rmag e imag, desvio para o vermelho (zph), logaritmo da massa da galáxia (logM) e logaritmo da taxa de formação estelar (logSFR).

1. Faça um gráfico de logFxs versus logFxh. Sobreponha a este gráfico as seguintes curvas de regressão, cada uma com cor diferente: OLS, regressão quantílica da mediana, estimador de Nadaraya-Watson e LOESS.
2. Represente a densidade do espaço logM versus logSFR usando um histograma bidimensional e curvas de contorno.

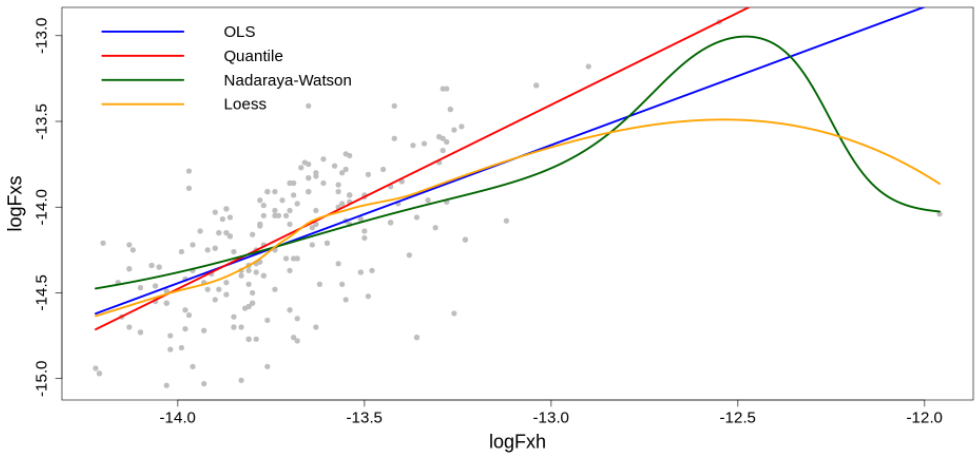

Figura 8: Dados do fluxo de raios-X moles vs duros. As linhas representam diferente tipos de regressões aplicadas nos dados.

Ao realizar uma regressão podemos escolher diferentes tipos de estimadores e sua escolha dependerá da natureza do problema. Neste cenário, vemos que a regressão local e a regressão usando o estimador de Nadaraya-Watson foram mais suscetíveis aos objetos nos extremos das distribuições, podendo gerar problemas de 'overfitting'.

```
# Data
agns = read.table('/content/xrays_tasse.tsv', header=T, sep='|')
```

```
agns = agns[complete.cases(agns),]

# Packages
install.packages('quantreg')
library(quantreg)
install.packages("np")
library(np)

# Fitting
fit_ols = lm(logFxs ~ logFxh, data = agns)
fit_qtl = rq(logFxs ~ logFxh, data = agns, tau = 0.5)
fit_nw = npreg(logFxs ~ logFxh, data = agns, bws = 0.2)
fit_loe = loess(logFxs ~ logFxh, data = agns)

# Points
x = seq(min(agns$logFxh), max(agns$logFxh), length.out = 200)
x_df = data.frame(logFxh = x)

# Plotting
options(repr.plot.width=16, repr.plot.height=8)
plot(agns$logFxh, agns$logFxs, pch=19, col='gray', xlab='logFxh', ylab='logFxs
    ',
 cex.lab=1.8, cex.axis=1.5)
lines(x, predict(fit_ols, newdata = x_df), col='blue', lwd=3)
lines(x, predict(fit_qtl, newdata = x_df), col='red', lwd=3)
lines(x, predict(fit_nw, exdat = x_df), col='darkgreen', lwd=3)
lines(x, predict(fit_loe, newdata = x_df), col='orange', lwd=3)
legend('topleft', legend=c('OLS', 'Quantile', 'Nadaraya-Watson', 'Loess'),
col=c('blue','red','darkgreen','orange'), lwd=3, cex=1.5, bty='n')
```

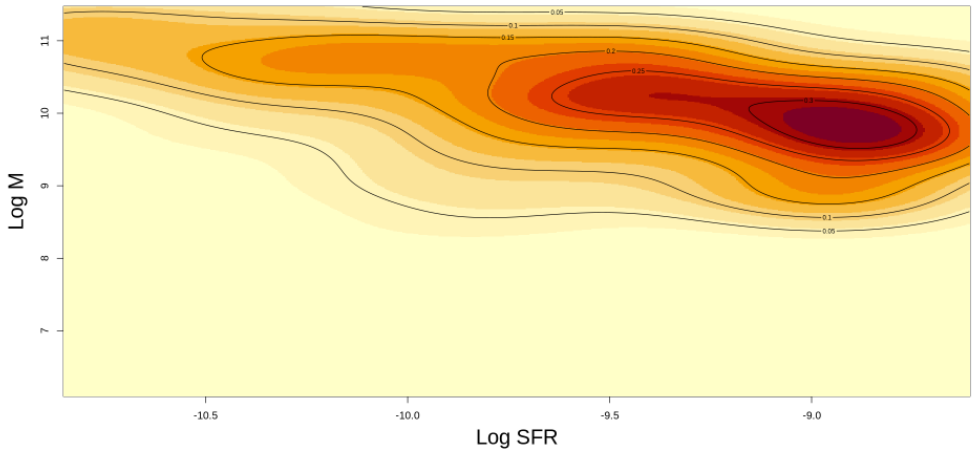

Figura 9: curvas de contorno com histograma 2D da massa da galáxia versus formação estelar, ambos em logaritmo.

b) Para representar a densidade bidimensional usei kernels gaussianos, onde cada ponto receberá uma distribuição gaussiana 2D com certa largura e ao final todas distribuições serão levadas em conta no histograma 2D. Percebe-se que galáxias com uma baixa taxa de formação estelar são geralmente mais massivas. Isso pode ser explicado pela evolução de galáxias, onde as galáxias mais massivas são elípticas vermelhas que

já estão velhas e sem formação de estrelas, enquanto as galáxias de menor massa possuem maior formação estelar e são mais jovens.

```
kde = kde2d(agns$logSFR, agns$logM, n=500)
image(kde, xlab='Log SFR', ylab='Log M', cex.lab=2)
contour(kde,add=T)
```

Questão 9 - O arquivo AsteroidClass.tsv contém os dados da fotometria de asteroides obtidos por Popescu et al. (2018). As colunas representam as cores YJ, JKs, HKs e a classificação espectroscópica do asteroide. Construa uma árvore de classificação para essa amostra, com base nas cores dos asteroides.

Para criar a árvore de classificação, utilizei as cores para explicar a classe espectroscópica. Então analisei o resultado através do sumário e do gráfico jitter 11. Percebe-se diversas classificações erradas, principalmente para objetos que realmente são da classe C. Possivelmente há poucos objetos dessa classe e estão em regiões de sobreposição no espaço multivariacional do problema. Para uma melhor performance, poderíamos usar florestas aleatórias, que é uma extensão de árvores de classificação.

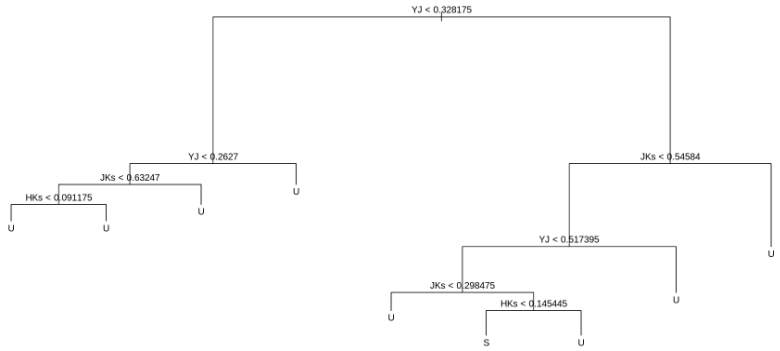

Figura 10: Árvore de classificação para a amostra de asteroides.

```
# Data
asteroid = read.table('/content/AsteroidClass.tsv', header=T, sep='|')
asteroid = asteroid[complete.cases(asteroid),]

# Packages
install.packages('tree')
library(tree)

# Creating tree
arv = tree(as.factor(Class) ~ HKs + JKs + YJ , data=asteroid)
summary(arv)

### output ###

Classification tree:
tree(formula = as.factor(Class) ~ HKs + JKs + YJ, data = asteroid)
Number of terminal nodes:  9
Residual mean deviance:  0.5786 = 4007 / 6926
Misclassification error rate: 0.1256 = 871 / 6935
```

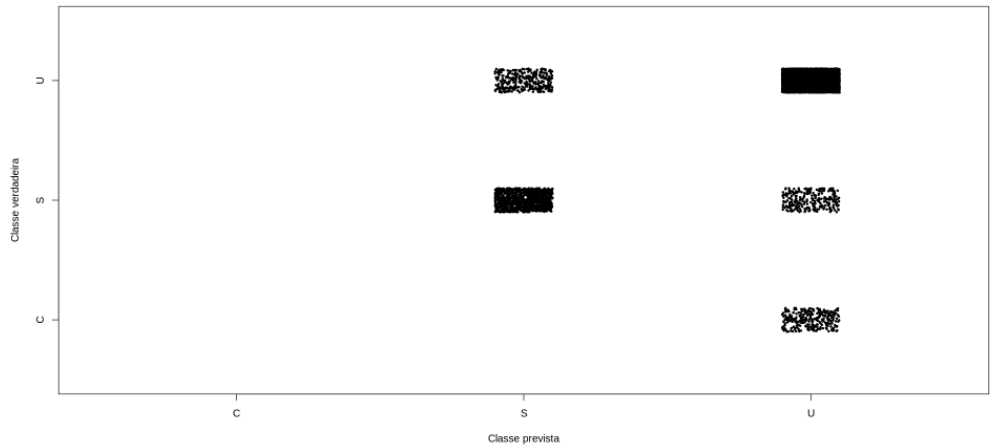

Figura 11: Gráfico jitter para o resultados da árvore de classificação aplicada na amostra de asteroides.

```
# Plot
plot(arv)
text(arv)

# Jitter plot
pred = predict(arv, type = "class")
asteroid$Class = as.factor(asteroid$Class)

plot(jitter(as.numeric(pred), factor=0.5),
     jitter(as.numeric(asteroid$Class), factor=0.5),
     pch=20, cex=0.6,
     xlab='Classe prevista', ylab='Classe verdadeira',
     xlim=c(0.5, 3.5),
     ylim=c(0.5, 3.5),
     axes=FALSE)

axis(1, at=1:3, labels=levels(asteroid$Class))
axis(2, at=1:3, labels=levels(asteroid$Class))
box()
```

Questão 10 - Gaspar et al. (2003) publicaram dados de fotometria de estrelas do aglomerado NGC 2126. Esses dados estão no arquivo ngc2126.dat. Nesta análise, use apenas as componentes de movimento próprio (pmRA e pmDE). Nem todas as estrelas dessa tabela são membros reais de NGC 2126; a rigor, a maioria não deve ser. Para a análise, considere que toda estrela com movimento próprio total nulo provavelmente são estrelas mais distantes e não pertencem ao aglomerado. Entre as estrelas restantes, ainda deve haver algumas intrusas que se encontrem entre nós e o aglomerado; contudo, como o aglomerado se move de forma coesa, ele possui um valor bem marcado em pmRA e pmDE, e as estrelas intrusas serão outliers na distribuição dessas componentes. Use um método de densidade por kernel bidimensional para representar a densidade das estrelas no espaço de componentes do movimento próprio. Estime o movimento próprio mais provável desse aglomerado a partir das coordenadas (pmRA, pmDE) de maior densidade no seu gráfico. Anote-as no gráfico com o símbolo + vermelho, em tamanho cex = 2.5.

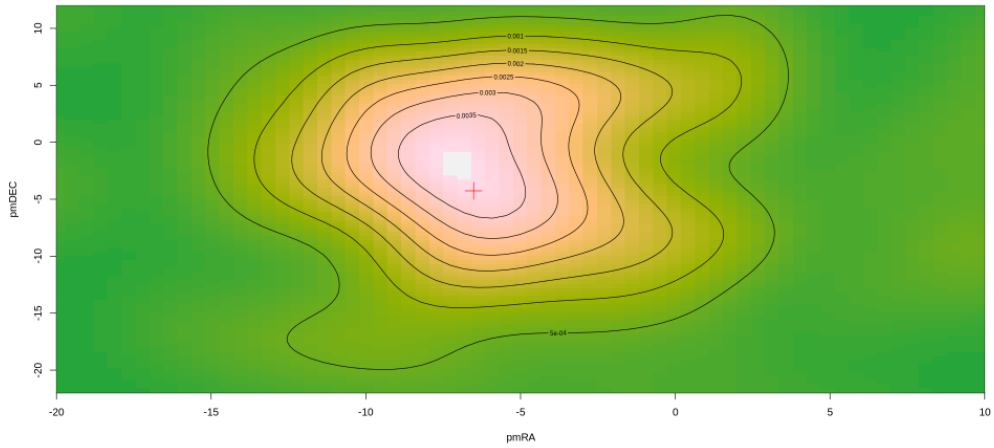

Figura 12: Curvas de contorno juntas ao histograma 2D por kenerls das componentes RA e DEC do movimento próprio das estrelas do algomerado. A cruz vermelha indica o valor mais provável para o movimento próprio do aglomerado após separação de misturas.

Para estimar a densidade bidimensional por kernel, utilizei o MASS:kde2d e o contour no kde para adicionar contornos. Como essa amostra contém outliers que não devem se mover da mesma maneira que o

aglomerado, decidi utilizar o `mclust` para separar as possíveis misturas da amostra. Após realizar a separação em duas componentes gaussianas, o valor mais provável de movimento do aglomerado será o máximo de uma das gaussianas, ou seja, sua média, como mostrado na figura 12.

```
# Data
ngc = read.table('/content/ngc2126.dat', sep='|', header=T)
mask = sqrt(ngc$pmDE**2+ngc$pmRA**2)>0
sample = ngc[mask,]

# Splitting mixture
model = Mclust(sample[,4:5],modelNames = 'VVV')
model$parameters$mean

### output ###
A matrix: 2    3 of type dbl
pmRA         -6.517683        -4.151874        -0.9629393
pmDE         -4.262831        4.439619         -13.9000586

# Plotting
kde = kde2d(sample$pmRA, sample$pmDE, n=200)
image(kde, xlab='pmRA', ylab='pmDEC',col = hcl.colors(100, "terrain"),
xlim=c(-20,10),ylim=c(-22,12))

#Cmap reference
#https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/image
contour(kde,add=T)

# Point
points(model$parameters$mean[1,1],model$parameters$mean[2,1],pch=3,
cex=2.5,col='red')
```