Agrupamento com K-means sobre dados Médicos

Marcos Vinícius dos Santos Ferreira

2018-04-16

Contents

Database Diabetes

Diabetes 130-US hospitals for years 1999-2008 Data Set

Estes dados foram preparados para analisar os fatores relacionados à readmissão, bem como outros resultados referentes aos pacientes com diabetes.

Informação do dataset

O conjunto de dados representa 10 anos (1999-2008) de atendimento clínico em 130 hospitais dos EUA e redes de distribuição integradas. Inclui mais de 50 atributos multivariados que representam os resultados do paciente e do hospital. O dataset contém 100000 instâncias. Informações foram extraídas do banco de dados para encontros que satisfizeram os seguintes critérios.

```
1. É um encontro de internação (internação hospitalar).
2. É um encontro diabético.
3. O tempo de internação foi de no mínimo 1 dia e no máximo 14 dias.
```

4. Testes laboratoriais foram realizados durante o encontro.
5. Medicamentos foram administrados durante o encontro.

Os dados contêm atributos como número do paciente, raça, gênero, idade, tipo de internação, tempo no hospital, especialidade médica do médico admitido, número de exames laboratoriais realizados, resultado do exame de HbA1c, diagnóstico, número de medicamentos, medicamentos diabéticos, número de pacientes ambulatoriais , internação e visitas de emergência no ano anterior à hospitalização, etc.

Dificuldades Aprensentadas:

- Heterogêneas e difíceis em termos de valores ausentes
- Registros incompletos ou inconsistentes
- Alta dimensionalidade, entendida pelo número de características e por sua complexidade.

Paper

Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records

- Impacto da Medida de HbA1c nas Taxas de Readmissão Hospitalar: Análise de 70.000 Registros de Pacientes com Base de Dados Clínicos.
- **Hipótesi:** Nossa hipótese é que a medida da HbA1c está associada a uma redução nas taxas de readmissão em indivíduos internados no hospital.

Minha Contribuição

Fazer o uso do aprendizado de máquina não supervisionado para identificar relação entre dados clínicos de diabetes e fornecer indícios de quais fatores influenciam mais na doença, para então descrever quantos níveis da doença podem existir entre os pacientes.

Objetivo

- Quais fatores influenciam e ou apontam indícios sobre o avanço ou cura da da diabete, e quantos níveis da doença podem ser descritos?

Metodos

- Usar o Aprendizado de Máquina não supervisionado com o algoritmo k-means para indentificar padrões que possam identificar padrões no dataset que evidenciam tais indícios.

Hipótese de quais fatores mais influenciam no diabetes(avanço da doença).

Atributo	Tipo	Valores Ausentes(%)
Idade	Nominal	0
Discharge disposition	Nominal	0
Admission source	Nominal	0

Atributo	Tipo	Valores Ausentes(%)
Time in hospital	Numeric	0
Medical specialty	Nominal	59
Number of lab procedures	Numeric	0
Number of procedures	Numeric	0
Number of medications	Numeric	0
Number of emergency visits	Numeric	0
Diagnosis 1	Nominal	0
Diagnosis 2	Nominal	0
Diagnosis 3	Nominal	1
Número de Diagnósticos	Numeric	0
Glucose serum test result	Nominal	1

Carregando os dados

```
# lendo o dataset
data.diabetes <- read.csv('../../data/dataset_diabetes/diabetic_data.csv')

# Visualizando o dataset
str(data.diabetes)
```

```
## 'data.frame':    101766 obs. of  50 variables:
##  $ encounter_id            : int  2278392 149190 64410 500364 16680 35754 55842 63768 12522 15738 ..
##  $ patient_nbr             : int  8222157 55629189 86047875 82442376 42519267 82637451 84259809 11488
##  $ race                    : Factor w/ 6 levels "?","AfricanAmerican",..: 4 4 2 4 4 4 4 4 4 4 ...
##  $ gender                  : Factor w/ 3 levels "Female","Male",..: 1 1 1 2 2 2 2 2 1 1 ...
##  $ age                     : Factor w/ 10 levels "[0-10)","[10-20)",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ weight                  : Factor w/ 10 levels "?","[0-25)","[100-125)",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ admission_type_id       : int  6 1 1 1 1 2 3 1 2 3 ...
##  $ discharge_disposition_id: int  25 1 1 1 1 1 1 1 1 3 ...
##  $ admission_source_id     : int  1 7 7 7 7 2 2 7 4 4 ...
##  $ time_in_hospital        : int  1 3 2 2 1 3 4 5 13 12 ...
##  $ payer_code              : Factor w/ 18 levels "?","BC","CH",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ medical_specialty       : Factor w/ 73 levels "?","AllergyandImmunology",..: 39 1 1 1 1 1 1 1 1 2
##  $ num_lab_procedures      : int  41 59 11 44 51 31 70 73 68 33 ...
##  $ num_procedures          : int  0 0 5 1 0 6 1 0 2 3 ...
##  $ num_medications         : int  1 18 13 16 8 16 21 12 28 18 ...
##  $ number_outpatient       : int  0 0 2 0 0 0 0 0 0 0 ...
##  $ number_emergency        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ number_inpatient        : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ diag_1                  : Factor w/ 717 levels "?","10","11",..: 126 145 456 556 56 265 265 278 25
##  $ diag_2                  : Factor w/ 749 levels "?","11","110",..: 1 81 80 99 26 248 248 316 262 48
##  $ diag_3                  : Factor w/ 790 levels "?","11","110",..: 1 123 768 250 88 88 772 88 231 3
##  $ number_diagnoses        : int  1 9 6 7 5 9 7 8 8 8 ...
##  $ max_glu_serum           : Factor w/ 4 levels ">200",">300",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ A1Cresult               : Factor w/ 4 levels ">7",">8","None",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ metformin               : Factor w/ 4 levels "Down","No","Steady",..: 2 2 2 2 2 2 3 2 2 2 ...
##  $ repaglinide             : Factor w/ 4 levels "Down","No","Steady",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ nateglinide             : Factor w/ 4 levels "Down","No","Steady",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ chlorpropamide          : Factor w/ 4 levels "Down","No","Steady",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ glimepiride             : Factor w/ 4 levels "Down","No","Steady",..: 2 2 2 2 2 2 3 2 2 2 ...
##  $ acetohexamide           : Factor w/ 2 levels "No","Steady": 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ glipizide               : Factor w/ 4 levels "Down","No","Steady",..: 2 2 3 2 3 2 2 2 3 2 ...
##  $ glyburide               : Factor w/ 4 levels "Down","No","Steady",..: 2 2 2 2 2 2 2 3 2 2 ...
##  $ tolbutamide             : Factor w/ 2 levels "No","Steady": 1 1 1 1 1 1 1 1 1 1 ...
##  $ pioglitazone            : Factor w/ 4 levels "Down","No","Steady",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ rosiglitazone           : Factor w/ 4 levels "Down","No","Steady",..: 2 2 2 2 2 2 2 2 2 3 ...
##  $ acarbose                : Factor w/ 4 levels "Down","No","Steady",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ miglitol                : Factor w/ 4 levels "Down","No","Steady",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ troglitazone            : Factor w/ 2 levels "No","Steady": 1 1 1 1 1 1 1 1 1 1 ...
##  $ tolazamide              : Factor w/ 3 levels "No","Steady",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ examide                 : Factor w/ 1 level "No": 1 1 1 1 1 1 1 1 1 1 ...
##  $ citoglipton             : Factor w/ 1 level "No": 1 1 1 1 1 1 1 1 1 1 ...
##  $ insulin                 : Factor w/ 4 levels "Down","No","Steady",..: 2 4 2 4 3 3 3 2 3 3 ...
##  $ glyburide.metformin     : Factor w/ 4 levels "Down","No","Steady",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ glipizide.metformin     : Factor w/ 2 levels "No","Steady": 1 1 1 1 1 1 1 1 1 1 ...
##  $ glimepiride.pioglitazone: Factor w/ 2 levels "No","Steady": 1 1 1 1 1 1 1 1 1 1 ...
##  $ metformin.rosiglitazone : Factor w/ 2 levels "No","Steady": 1 1 1 1 1 1 1 1 1 1 ...
##  $ metformin.pioglitazone  : Factor w/ 2 levels "No","Steady": 1 1 1 1 1 1 1 1 1 1 ...
##  $ change                  : Factor w/ 2 levels "Ch","No": 2 1 2 1 1 2 1 2 1 1 ...
##  $ diabetesMed             : Factor w/ 2 levels "No","Yes": 1 2 2 2 2 2 2 2 2 2 ...
##  $ readmitted              : Factor w/ 3 levels "<30",">30","NO": 3 2 3 3 3 2 3 2 3 3 ...
```

Pré-Processamento

```
# selecionar os atributos que por hipótese podem ser relevantes
data.diabetes.rel <- data.diabetes[,c('age', 'discharge_disposition_id', 'admission_source_id',
                                      'time_in_hospital', 'medical_specialty', 'num_lab_procedures', 'nu
                                      'num_medications', 'number_outpatient', 'diag_1', 'diag_2', 'diag_
                                      'number_diagnoses', 'max_glu_serum')]

# estrutura dos novos dados
str(data.diabetes.rel)
```

```
## 'data.frame':    101766 obs. of  14 variables:
##  $ age                     : Factor w/ 10 levels "[0-10)","[10-20)",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ discharge_disposition_id: int  25 1 1 1 1 1 1 1 1 3 ...
##  $ admission_source_id     : int  1 7 7 7 7 2 2 7 4 4 ...
##  $ time_in_hospital        : int  1 3 2 2 1 3 4 5 13 12 ...
##  $ medical_specialty       : Factor w/ 73 levels "?","AllergyandImmunology",..: 39 1 1 1 1 1 1 1 1 20
##  $ num_lab_procedures      : int  41 59 11 44 51 31 70 73 68 33 ...
##  $ num_procedures          : int  0 0 5 1 0 6 1 0 2 3 ...
##  $ num_medications         : int  1 18 13 16 8 16 21 12 28 18 ...
##  $ number_outpatient       : int  0 0 2 0 0 0 0 0 0 0 ...
##  $ diag_1                  : Factor w/ 717 levels "?","10","11",..: 126 145 456 556 56 265 265 278 25
##  $ diag_2                  : Factor w/ 749 levels "?","11","110",..: 1 81 80 99 26 248 248 316 262 48
##  $ diag_3                  : Factor w/ 790 levels "?","11","110",..: 1 123 768 250 88 88 772 88 231 3
##  $ number_diagnoses        : int  1 9 6 7 5 9 7 8 8 8 ...
##  $ max_glu_serum           : Factor w/ 4 levels ">200",">300",..: 3 3 3 3 3 3 3 3 3 3 ...
```

```
# selecionar as 200 primeiras instancias

data.diabetes.rel.sel <- data.diabetes.rel[1:200,]

str(data.diabetes.rel.sel)
```

```
## 'data.frame':    200 obs. of  14 variables:
##  $ age                     : Factor w/ 10 levels "[0-10)","[10-20)",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ discharge_disposition_id: int  25 1 1 1 1 1 1 1 1 3 ...
##  $ admission_source_id     : int  1 7 7 7 7 2 2 7 4 4 ...
##  $ time_in_hospital        : int  1 3 2 2 1 3 4 5 13 12 ...
##  $ medical_specialty       : Factor w/ 73 levels "?","AllergyandImmunology",..: 39 1 1 1 1 1 1 1 1 20
##  $ num_lab_procedures      : int  41 59 11 44 51 31 70 73 68 33 ...
##  $ num_procedures          : int  0 0 5 1 0 6 1 0 2 3 ...
##  $ num_medications         : int  1 18 13 16 8 16 21 12 28 18 ...
##  $ number_outpatient       : int  0 0 2 0 0 0 0 0 0 0 ...
##  $ diag_1                  : Factor w/ 717 levels "?","10","11",..: 126 145 456 556 56 265 265 278 25
##  $ diag_2                  : Factor w/ 749 levels "?","11","110",..: 1 81 80 99 26 248 248 316 262 48
##  $ diag_3                  : Factor w/ 790 levels "?","11","110",..: 1 123 768 250 88 88 772 88 231 3
##  $ number_diagnoses        : int  1 9 6 7 5 9 7 8 8 8 ...
##  $ max_glu_serum           : Factor w/ 4 levels ">200",">300",..: 3 3 3 3 3 3 3 3 3 3 ...
```

```
# tranformar as colunas de palavras em numeros
med.esp <-as.numeric(data.diabetes.rel.sel$medical_specialty)

data.diabetes.rel.sel['medical_specialty'] <- as.integer(med.esp)

str(data.diabetes.rel.sel)
```

```
## 'data.frame':    200 obs. of  14 variables:
##  $ age                     : Factor w/ 10 levels "[0-10)","[10-20)",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ discharge_disposition_id: int  25 1 1 1 1 1 1 1 1 3 ...
##  $ admission_source_id     : int  1 7 7 7 7 2 2 7 4 4 ...
##  $ time_in_hospital        : int  1 3 2 2 1 3 4 5 13 12 ...
##  $ medical_specialty       : int  39 1 1 1 1 1 1 1 1 20 ...
##  $ num_lab_procedures      : int  41 59 11 44 51 31 70 73 68 33 ...
##  $ num_procedures          : int  0 0 5 1 0 6 1 0 2 3 ...
##  $ num_medications         : int  1 18 13 16 8 16 21 12 28 18 ...
##  $ number_outpatient       : int  0 0 2 0 0 0 0 0 0 0 ...
##  $ diag_1                  : Factor w/ 717 levels "?","10","11",..: 126 145 456 556 56 265 265 278 25
##  $ diag_2                  : Factor w/ 749 levels "?","11","110",..: 1 81 80 99 26 248 248 316 262 48
##  $ diag_3                  : Factor w/ 790 levels "?","11","110",..: 1 123 768 250 88 88 772 88 231 3
##  $ number_diagnoses        : int  1 9 6 7 5 9 7 8 8 8 ...
##  $ max_glu_serum           : Factor w/ 4 levels ">200",">300",..: 3 3 3 3 3 3 3 3 3 3 ...
```

```
# dados de idade

data.diabetes.rel.sel$age
```

```
##   [1] [0-10)   [10-20)  [20-30)  [30-40)  [40-50)  [50-60)  [60-70)
##   [8] [70-80)  [80-90)  [90-100) [40-50)  [60-70)  [40-50)  [80-90)
##  [15] [60-70)  [60-70)  [50-60)  [50-60)  [70-80)  [70-80)  [50-60)
##  [22] [60-70)  [70-80)  [80-90)  [70-80)  [50-60)  [80-90)  [50-60)
##  [29] [20-30)  [80-90)  [60-70)  [70-80)  [70-80)  [60-70)  [70-80)
##  [36] [60-70)  [70-80)  [60-70)  [70-80)  [50-60)  [70-80)  [40-50)
##  [43] [70-80)  [50-60)  [80-90)  [40-50)  [70-80)  [70-80)  [50-60)
##  [50] [60-70)  [50-60)  [70-80)  [40-50)  [50-60)  [60-70)  [60-70)
##  [57] [50-60)  [40-50)  [80-90)  [70-80)  [70-80)  [50-60)  [40-50)
##  [64] [80-90)  [50-60)  [90-100) [10-20)  [80-90)  [50-60)  [50-60)
##  [71] [70-80)  [50-60)  [60-70)  [70-80)  [70-80)  [70-80)  [60-70)
##  [78] [60-70)  [50-60)  [50-60)  [70-80)  [50-60)  [50-60)  [60-70)
##  [85] [60-70)  [40-50)  [40-50)  [60-70)  [60-70)  [40-50)  [70-80)
```

```
##  [92] [70-80)  [40-50)  [50-60)  [60-70)  [70-80)  [70-80)  [70-80)
##  [99] [50-60)  [30-40)  [70-80)  [60-70)  [30-40)  [60-70)  [70-80)
## [106] [80-90)  [50-60)  [80-90)  [60-70)  [50-60)  [50-60)  [60-70)
## [113] [40-50)  [70-80)  [70-80)  [30-40)  [60-70)  [70-80)  [60-70)
## [120] [60-70)  [70-80)  [40-50)  [40-50)  [70-80)  [50-60)  [30-40)
## [127] [80-90)  [30-40)  [20-30)  [60-70)  [50-60)  [60-70)  [60-70)
## [134] [70-80)  [90-100) [70-80)  [60-70)  [60-70)  [50-60)  [80-90)
## [141] [30-40)  [60-70)  [80-90)  [20-30)  [90-100) [50-60)  [50-60)
## [148] [50-60)  [50-60)  [70-80)  [60-70)  [40-50)  [50-60)  [70-80)
## [155] [50-60)  [60-70)  [60-70)  [50-60)  [60-70)  [80-90)  [80-90)
## [162] [50-60)  [80-90)  [50-60)  [80-90)  [40-50)  [80-90)  [30-40)
## [169] [50-60)  [50-60)  [60-70)  [60-70)  [70-80)  [50-60)  [70-80)
## [176] [70-80)  [80-90)  [40-50)  [70-80)  [40-50)  [40-50)  [70-80)
## [183] [50-60)  [50-60)  [60-70)  [70-80)  [80-90)  [40-50)  [40-50)
## [190] [70-80)  [70-80)  [20-30)  [40-50)  [60-70)  [20-30)  [60-70)
## [197] [60-70)  [40-50)  [30-40)  [20-30)
## 10 Levels: [0-10) [10-20) [20-30) [30-40) [40-50) [50-60) ... [90-100)
```

Tratando os valores de diagnostico.

```
# função que troca de valores
troca.valor<-function(estrutura, valor, troca){

  temp <- as.vector(estrutura)

  temp[which(temp==valor)]=troca

  temp
}

data.diabetes.rel.sel$age <- troca.valor(data.diabetes.rel.sel$age,"[0-10)", "10")
data.diabetes.rel.sel$age <- troca.valor(data.diabetes.rel.sel$age,"[10-20)", "15")
data.diabetes.rel.sel$age <- troca.valor(data.diabetes.rel.sel$age,"[20-30)", "25")
data.diabetes.rel.sel$age <- troca.valor(data.diabetes.rel.sel$age,"[30-40)", "35")
data.diabetes.rel.sel$age <- troca.valor(data.diabetes.rel.sel$age,"[40-50)", "45")
data.diabetes.rel.sel$age <- troca.valor(data.diabetes.rel.sel$age,"[50-60)", "55")
data.diabetes.rel.sel$age <- troca.valor(data.diabetes.rel.sel$age,"[60-70)", "65")
data.diabetes.rel.sel$age <- troca.valor(data.diabetes.rel.sel$age,"[70-80)", "75")
data.diabetes.rel.sel$age <- troca.valor(data.diabetes.rel.sel$age,"[80-90)", "85")
data.diabetes.rel.sel$age <- troca.valor(data.diabetes.rel.sel$age,"[90-100)", "95")

# vendoos valores convertidos pela media de idades
data.diabetes.rel.sel$age
```

```
##   [1] "10" "15" "25" "35" "45" "55" "65" "75" "85" "95" "45" "65" "45" "85"
##  [15] "65" "65" "55" "55" "75" "75" "55" "65" "75" "85" "75" "55" "85" "55"
##  [29] "25" "85" "65" "75" "75" "65" "75" "65" "75" "65" "75" "55" "75" "45"
##  [43] "75" "55" "85" "45" "75" "75" "55" "65" "55" "75" "45" "55" "65" "65"
##  [57] "55" "45" "85" "75" "75" "55" "45" "85" "55" "95" "15" "85" "55" "55"
##  [71] "75" "55" "65" "75" "75" "75" "65" "65" "55" "55" "75" "55" "55" "65"
##  [85] "65" "45" "45" "65" "65" "45" "75" "75" "45" "55" "65" "75" "75" "75"
##  [99] "55" "35" "75" "65" "35" "65" "75" "85" "55" "85" "65" "55" "55" "65"
## [113] "45" "75" "75" "35" "65" "75" "65" "65" "75" "45" "45" "75" "55" "35"
## [127] "85" "35" "25" "65" "55" "65" "65" "75" "95" "75" "65" "65" "55" "85"
## [141] "35" "65" "85" "25" "95" "55" "55" "55" "55" "75" "65" "45" "55" "75"
```

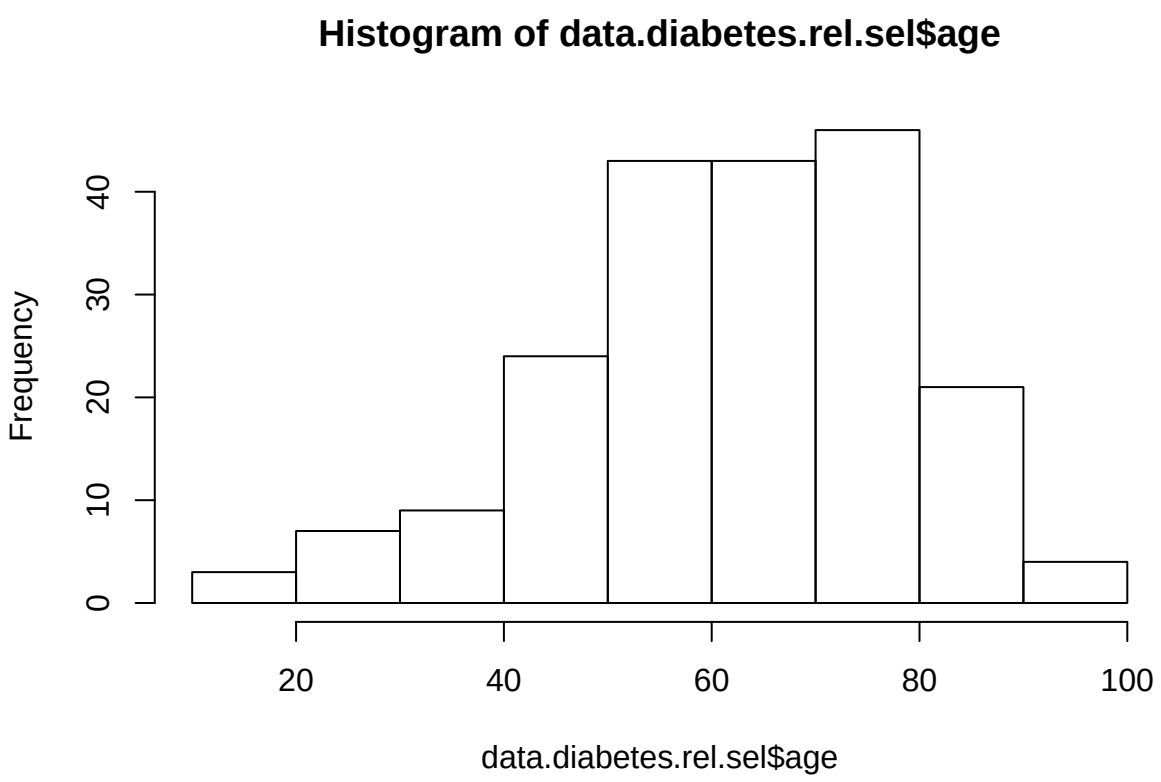
```
## [155] "55" "65" "65" "55" "65" "85" "85" "55" "85" "55" "85" "45" "85" "35"
## [169] "55" "55" "65" "65" "75" "55" "75" "75" "85" "45" "75" "45" "45" "75"
## [183] "55" "55" "65" "75" "85" "45" "45" "75" "75" "25" "45" "65" "25" "65"
## [197] "65" "45" "35" "25"
```

```
# convertendo os nuvemos para interios
data.diabetes.rel.sel$age <- as.integer(data.diabetes.rel.sel$age)

data.diabetes.rel.sel$age
```

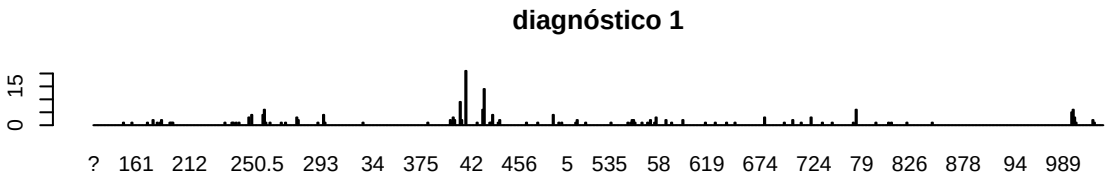
```
##   [1] 10 15 25 35 45 55 65 75 85 95 45 65 45 85 65 65 55 55 75 75 55 65 75
##  [24] 85 75 55 85 55 25 85 65 75 75 65 75 65 75 65 75 55 75 45 75 55 85 45
##  [47] 75 75 55 65 55 75 45 55 65 65 55 45 85 75 75 55 45 85 55 95 15 85 55
##  [70] 55 75 55 65 75 75 75 65 65 55 55 75 55 55 65 65 45 45 65 65 45 75 75
##  [93] 45 55 65 75 75 75 55 35 75 65 35 65 75 85 55 85 65 55 55 65 45 75 75
## [116] 35 65 75 65 65 75 45 45 75 55 35 85 35 25 65 55 65 65 75 95 75 65 65
## [139] 55 85 35 65 85 25 95 55 55 55 55 75 65 45 55 75 55 65 65 55 65 85 85
## [162] 55 85 55 85 45 85 35 55 55 65 65 75 55 75 75 85 45 75 45 45 75 55 55
## [185] 65 75 85 45 45 75 75 25 45 65 25 65 65 45 35 25
```

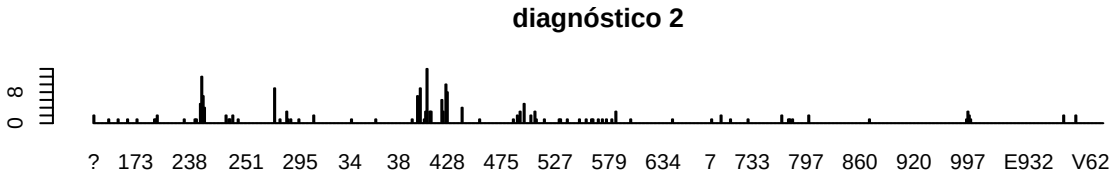
vendo a distribuição dos valores com o histograma.

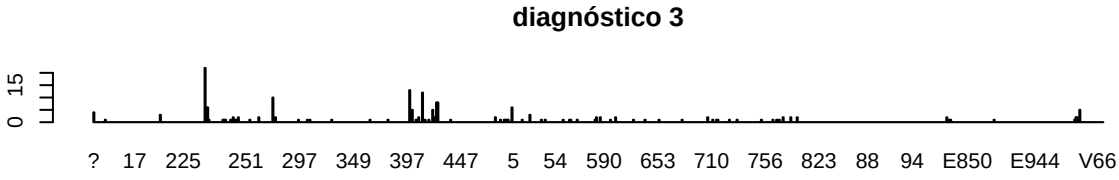
```
hist(data.diabetes.rel.sel$age)
```




```
# tratar valores nulos dos diagnóstico - solução - média.

# visualizando os dados dos 3 dianosticos
par(mfrow=c(3,1))
plot(data.diabetes.rel.sel$diag_1, main='diagnóstico 1')
plot(data.diabetes.rel.sel$diag_2, main='diagnóstico 2')
plot(data.diabetes.rel.sel$diag_3, main='diagnóstico 3')
```





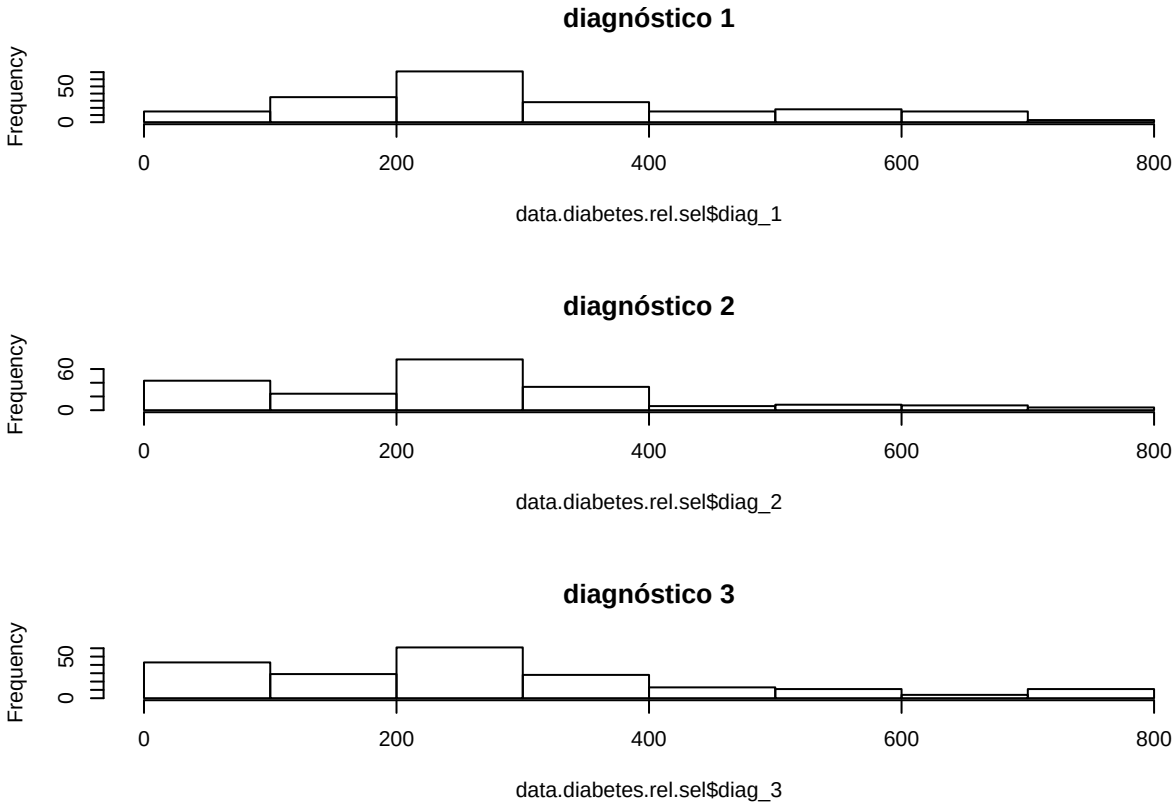



```
#tratar valores desconhecidos - ?
## possso remover esses valores ou aplicar um media
head(data.diabetes.rel.sel$diag_1)
```

```
## [1] 250.83 276    648    8      197    414
## 717 Levels: ? 10 11 110 112 114 115 117 131 133 135 136 141 142 143 ... V71
```

```
# solution 1
# data.diabetes.rel.sel$diag_1 <- troca.valor(data.diabetes.rel.sel$diag_1,'?', 414)
# data.diabetes.rel.sel$diag_1 <- as.integer(data.diabetes.rel.sel$diag_1)
# data.diabetes.rel.sel$diag_1[is.na(data.diabetes.rel.sel$diag_1)] <- 0
# data.diabetes.rel.sel$diag_1

# solution 2
# temp <- troca.valor(data.diabetes.rel.sel$diag_1,'?', 414)
# temp <- as.integer(temp)
# temp[is.na(temp)] <- 0
# hist(temp)

# solution 3
# diagnostico 1
data.diabetes.rel.sel$diag_1 <- as.integer(data.diabetes.rel.sel$diag_1)
data.diabetes.rel.sel$diag_1[is.na(data.diabetes.rel.sel$diag_1)] <- 0

# diagnostico 2
data.diabetes.rel.sel$diag_2 <- as.integer(data.diabetes.rel.sel$diag_2)
data.diabetes.rel.sel$diag_2[is.na(data.diabetes.rel.sel$diag_2)] <- 0

# diagnostico 3
data.diabetes.rel.sel$diag_3 <- as.integer(data.diabetes.rel.sel$diag_3)
data.diabetes.rel.sel$diag_3[is.na(data.diabetes.rel.sel$diag_3)] <- 0
```

```r
# visualizando os dados
par(mfrow=c(3,1))
hist(data.diabetes.rel.sel$diag_1, main='diagnóstico 1')
hist(data.diabetes.rel.sel$diag_2, main='diagnóstico 2')
hist(data.diabetes.rel.sel$diag_3, main='diagnóstico 3')
```




```r
# conferindo a nova estrutura do dataset
str(data.diabetes.rel.sel)
```

```
## 'data.frame':    200 obs. of  14 variables:
##  $ age                     : int  10 15 25 35 45 55 65 75 85 95 ...
##  $ discharge_disposition_id: int  25 1 1 1 1 1 1 1 1 3 ...
##  $ admission_source_id     : int  1 7 7 7 7 2 2 7 4 4 ...
##  $ time_in_hospital        : int  1 3 2 2 1 3 4 5 13 12 ...
##  $ medical_specialty       : int  39 1 1 1 1 1 1 1 1 20 ...
##  $ num_lab_procedures      : int  41 59 11 44 51 31 70 73 68 33 ...
##  $ num_procedures          : int  0 0 5 1 0 6 1 0 2 3 ...
##  $ num_medications         : int  1 18 13 16 8 16 21 12 28 18 ...
##  $ number_outpatient       : int  0 0 2 0 0 0 0 0 0 0 ...
##  $ diag_1                  : num  126 145 456 556 56 265 265 278 254 284 ...
##  $ diag_2                  : num  1 81 80 99 26 248 248 316 262 48 ...
##  $ diag_3                  : num  1 123 768 250 88 88 772 88 231 319 ...
##  $ number_diagnoses        : int  1 9 6 7 5 9 7 8 8 8 ...
##  $ max_glu_serum           : Factor w/ 4 levels ">200",">300",..: 3 3 3 3 3 3 3 3 3 3 ...
```

```r
# removendo a coluna max_glu_serum por ter muitos valores discrepantes
data.diabetes.rel.sel = subset(data.diabetes.rel.sel, select = -c(max_glu_serum))
```

Análise dos dados

Primeiro temos que explorar e visualizar os dados.

```
# estrutura dos meus dados
str(data.diabetes.rel.sel)
```

```
## 'data.frame':    200 obs. of  13 variables:
##  $ age                     : int  10 15 25 35 45 55 65 75 85 95 ...
##  $ discharge_disposition_id: int  25 1 1 1 1 1 1 1 1 3 ...
##  $ admission_source_id     : int  1 7 7 7 7 2 2 7 4 4 ...
##  $ time_in_hospital        : int  1 3 2 2 1 3 4 5 13 12 ...
##  $ medical_specialty       : int  39 1 1 1 1 1 1 1 1 20 ...
##  $ num_lab_procedures      : int  41 59 11 44 51 31 70 73 68 33 ...
##  $ num_procedures          : int  0 0 5 1 0 6 1 0 2 3 ...
##  $ num_medications         : int  1 18 13 16 8 16 21 12 28 18 ...
##  $ number_outpatient       : int  0 0 2 0 0 0 0 0 0 0 ...
##  $ diag_1                  : num  126 145 456 556 56 265 265 278 254 284 ...
##  $ diag_2                  : num  1 81 80 99 26 248 248 316 262 48 ...
##  $ diag_3                  : num  1 123 768 250 88 88 772 88 231 319 ...
##  $ number_diagnoses        : int  1 9 6 7 5 9 7 8 8 8 ...
```

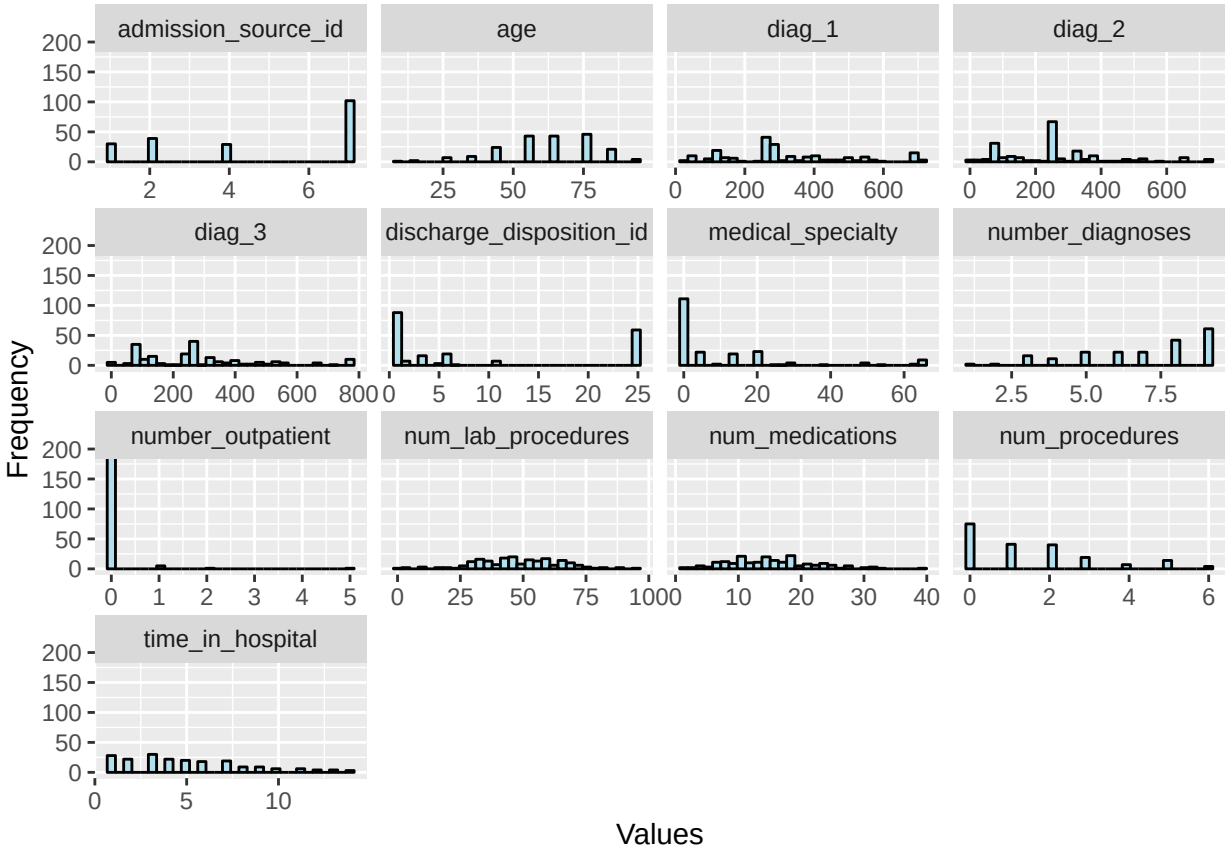
Todas as colunas são expressas como numéricas ou inteiras. E quanto à distribuição estatística?

```
summary(data.diabetes.rel.sel)
```

```
##       age        discharge_disposition_id admission_source_id
##  Min.   :10.00   Min.   : 1.00            Min.   :1.00
##  1st Qu.:55.00   1st Qu.: 1.00            1st Qu.:2.00
##  Median :65.00   Median : 3.00            Median :7.00
##  Mean   :61.92   Mean   : 9.19            Mean   :4.69
##  3rd Qu.:75.00   3rd Qu.:25.00            3rd Qu.:7.00
##  Max.   :95.00   Max.   :25.00            Max.   :7.00
##  time_in_hospital medical_specialty num_lab_procedures num_procedures
##  Min.   : 1.00    Min.   : 1.00     Min.   : 1.00      Min.   :0.0
##  1st Qu.: 2.75    1st Qu.: 1.00     1st Qu.:36.00      1st Qu.:0.0
##  Median : 4.00    Median : 1.00     Median :47.00      Median :1.0
##  Mean   : 5.06    Mean   :10.58     Mean   :48.17      Mean   :1.5
##  3rd Qu.: 7.00    3rd Qu.:13.25     3rd Qu.:59.00      3rd Qu.:2.0
##  Max.   :14.00    Max.   :66.00     Max.   :96.00      Max.   :6.0
##  num_medications number_outpatient     diag_1          diag_2
##  Min.   : 1.00   Min.   :0.00      Min.   : 22.0   Min.   :  1.0
##  1st Qu.:10.00   1st Qu.:0.00      1st Qu.:226.5   1st Qu.:135.0
##  Median :15.00   Median :0.00      Median :278.0   Median :248.0
##  Mean   :15.19   Mean   :0.06      Mean   :319.2   Mean   :255.6
##  3rd Qu.:19.00   3rd Qu.:0.00      3rd Qu.:407.0   3rd Qu.:320.0
##  Max.   :39.00   Max.   :5.00      Max.   :711.0   Max.   :729.0
##      diag_3      number_diagnoses
##  Min.   :  1.0   Min.   :1.000
##  1st Qu.:111.8   1st Qu.:5.000
##  Median :258.0   Median :8.000
##  Mean   :277.7   Mean   :6.895
##  3rd Qu.:344.2   3rd Qu.:9.000
##  Max.   :772.0   Max.   :9.000
```

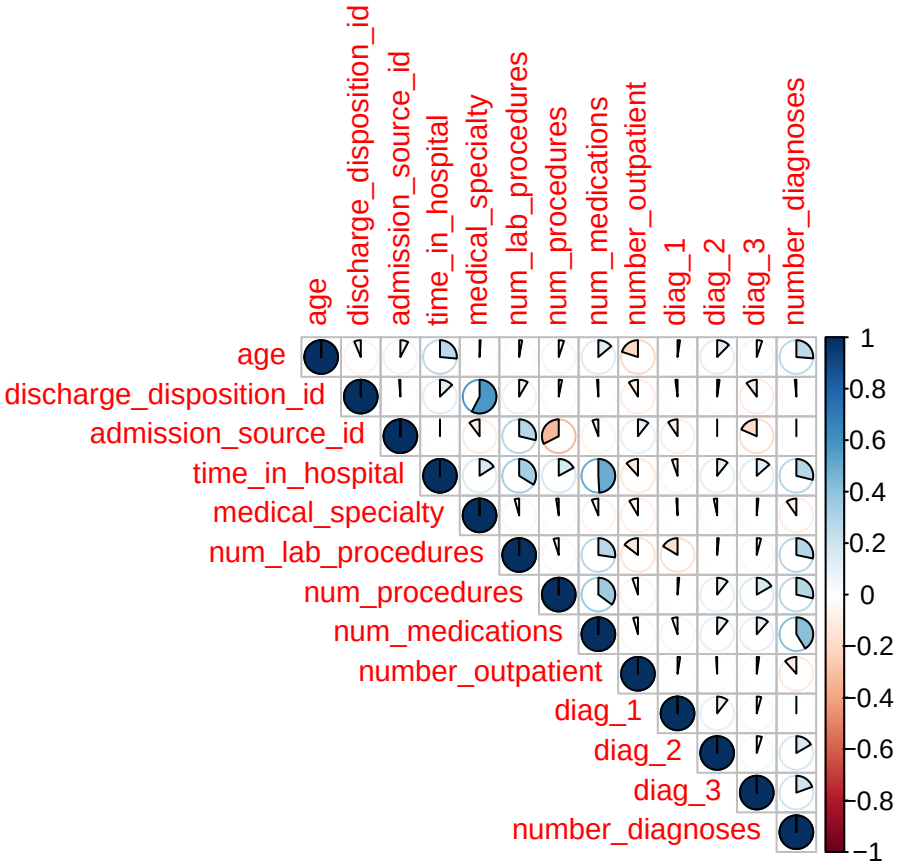
```
#load library
library(tidyverse)
```

```
library(corrplot)
library(gridExtra)
library(GGally)

# Histograma de cada atributo

data.diabetes.rel.sel %>%
  gather(Attributes, value, 1:13) %>%
  ggplot(aes(x=value)) +
  geom_histogram(fill="lightblue2", colour="black") +
  facet_wrap(~Attributes, scales="free_x") +
  labs(x="Values", y="Frequency")
```



Qual é a relação entre os diferentes atributos? Podemos usar a função **corrplot()** para criar uma exibição gráfica de uma matriz de correlação.

```
# Matrix de correlação - pie(boa representação)
corrplot(cor(data.diabetes.rel.sel), type="upper", method="pie", tl.cex=0.9)
```



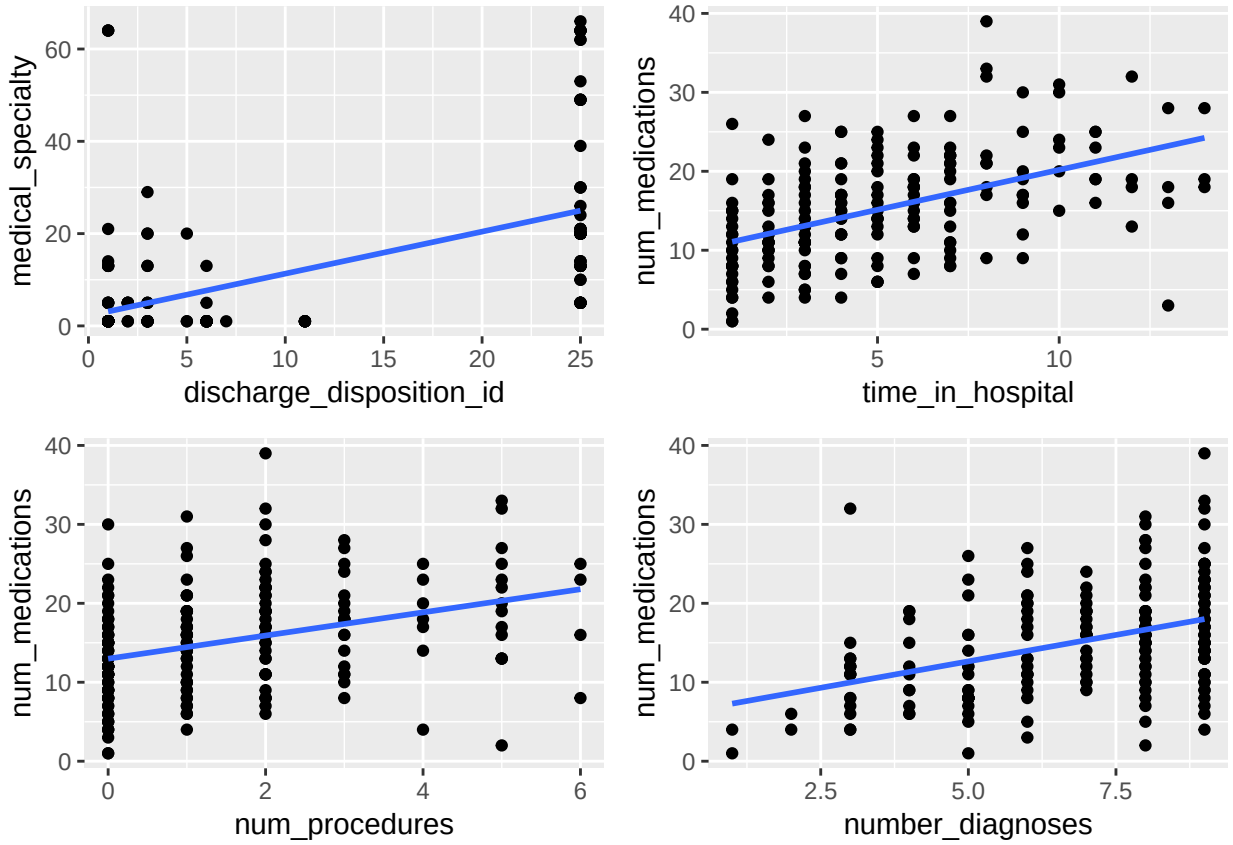
Existe uma forte correlação linear entre os atributos:

- *discharge-disposition-id* e *medical-specialty*
- *num-medications* e *time-in-hospital*
- *num-medications* e *num-procedures*
- *num-medications* e *number-diagnoses*
- *admission-source-id* e *num-procedures*

Podemos modelar a relação entre essas duas variáveis ajustando uma equação linear.

```
# Relacionamento entre as variaveis que mais tem correlação

# discrarge disposition id & médico especialista
plt1 <- ggplot(data.diabetes.rel.sel, aes(x=discharge_disposition_id, y=medical_specialty)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)

# tempo em hospital & numero de medicamentos
plt2 <- ggplot(data.diabetes.rel.sel, aes(x=time_in_hospital, y=num_medications)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)

# número de procedimentos e numero de medicamentos medications
plt3 <- ggplot(data.diabetes.rel.sel, aes(x=num_procedures, y=num_medications)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)

# numbero de diagnosticos e numero de medicações
```

```
plt4 <- ggplot(data.diabetes.rel.sel, aes(x=number_diagnoses, y=num_medications)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)

grid.arrange(plt1, plt2, plt3, plt4,  ncol=2, nrow=2)
```




```
#grid.arrange(plt1, plt2, plt3, plt4, nrow=2 , col=2)
```

A correlação linear se aplica fortemente entre as variáveis *num-medications*, *time-in-hospital*, *discharge-disposition-id*, *medical-specialty*, *num-procedures*, *number-diagnoses*, e *admission-source-id*. **Esses atributos fortemene ligados, podem nos levar a inferir descrições sobre os dados, ou seja, podemos conferir por exemplo a hipótesi de quantos níveis de diabetes as pessoas que estão realizando esse tratamento possuem.**

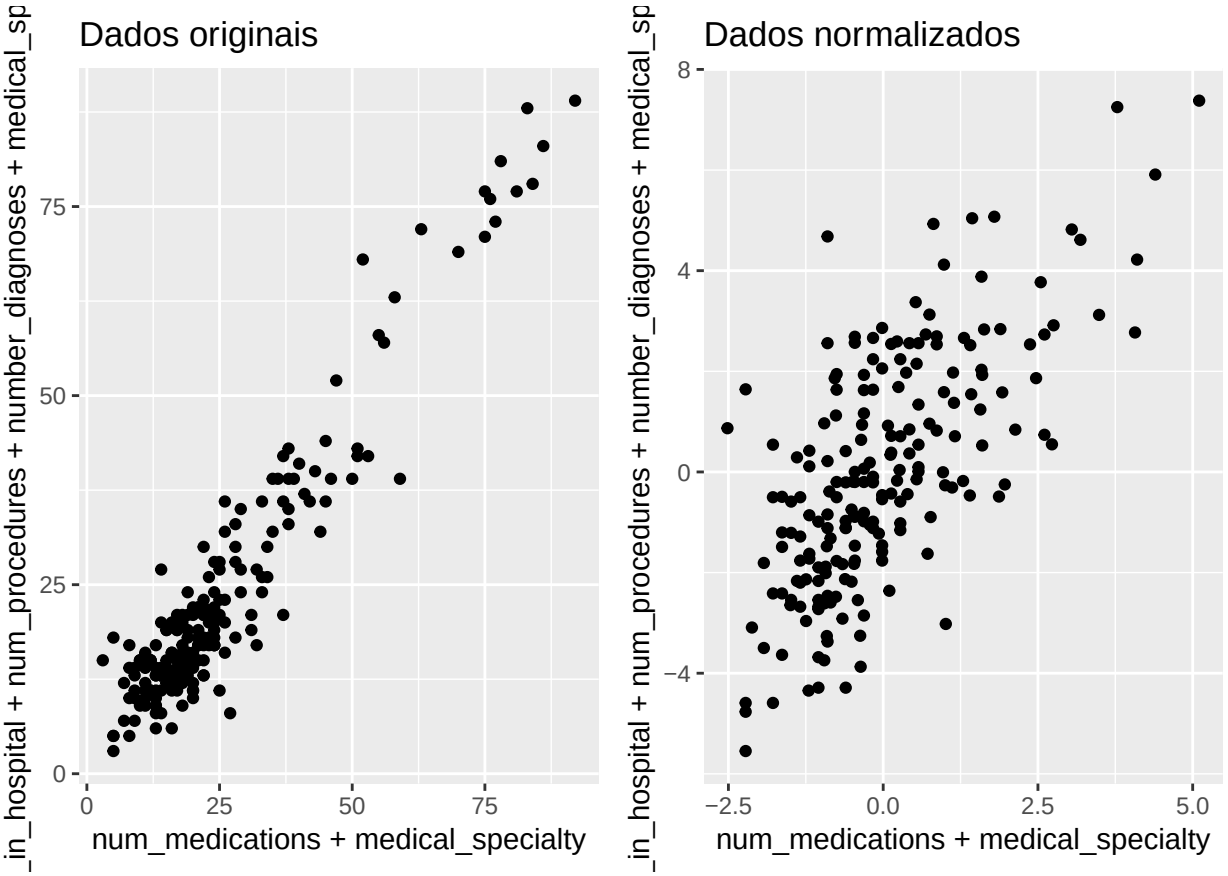
Agora que fizemos uma análise de dados exploratória, podemos descrever os dados ao encontrar grupos que compartilham padrões semelhantes, para tal, utilizaremos o algoritmo particional **K-means**, segundo mais importante na área de mineração de dados. Podemos preparar os dados para executar o algoritmo k-means.

Preparação dos dados

Temos que normalizar as variáveis para expressá-las no mesmo intervalo de valores. Em outras palavras, normalização significa ajustar os valores medidos em diferentes escalas para uma escala comum.

```
# após analisar os atributos que podem possuem maior relação, os mesmos
# são selecionados para inferir melhor os resultados.

# criar uma nova variavel para receber as novas
```

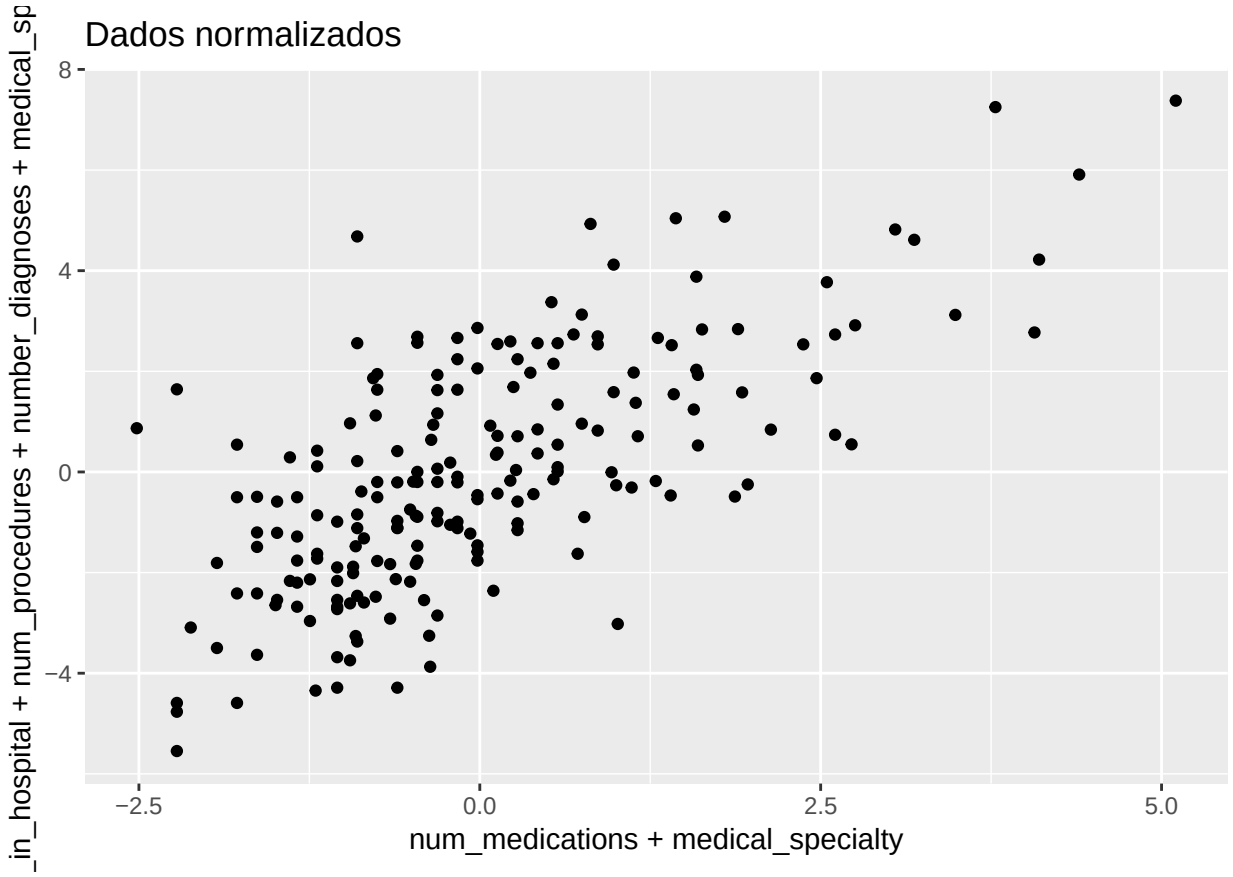
```
  # ------- dados que serao utilizados --------
  #  discharge_disposition_id *
  #  medical_specialty **
  #  num_medications
  #  time_in_hospital
  #  num_procedures
  #  number_diagnoses
  #  admission_source_id

# selecionando os atributos que podem inferir hipóteses relevantes
data.diabetes.rel.sel.new <-data.diabetes.rel.sel[,c('discharge_disposition_id', 'admission_source_id',
                                     'time_in_hospital', 'medical_specialty', 'num_procedures',
                                     'num_medications' , 'number_diagnoses')]

# atributos selecionados
data.diabetes.rel.sel.new.norm <- as.data.frame(scale(data.diabetes.rel.sel.new))


# dados originais sem normalização
data <- ggplot(data.diabetes.rel.sel.new,
               aes(x=num_medications + medical_specialty  ,
                   y=time_in_hospital + num_procedures + number_diagnoses + medical_specialty)) +
  geom_point() +
  labs(title="Dados originais")

# dados normalizados
data.norm <- ggplot(data.diabetes.rel.sel.new.norm,
               aes(x=num_medications + medical_specialty,
                   y=time_in_hospital + num_procedures + number_diagnoses + medical_specialty)) +
  geom_point() +
  labs(title="Dados normalizados")

# subplot
grid.arrange(data,data.norm, ncol=2)
```



Quantos Clustes ?

Para o algoritmo K-means encontrar similiaridade entre os grupos, precisa do parâmetro **k**, que é a quantidade de grupos a serem escolhidos. Utiliza-se o método do cotovelo que testa a variância dos dados em relação ao número de clusters. Ele testa até o momento que conforme o número de clustes aumenta não representa um valor significativo de ganho. Podemos ver o formato de um cotovelo ao plotar os resultados em uma gráfico e partir do valor indicado pelo *cotovelo* no gráfico significa que não existe ganho em relação ao aumento de clusters. Antes mesmo, visualizamos o conjunto de dados selecionado ate o momento.

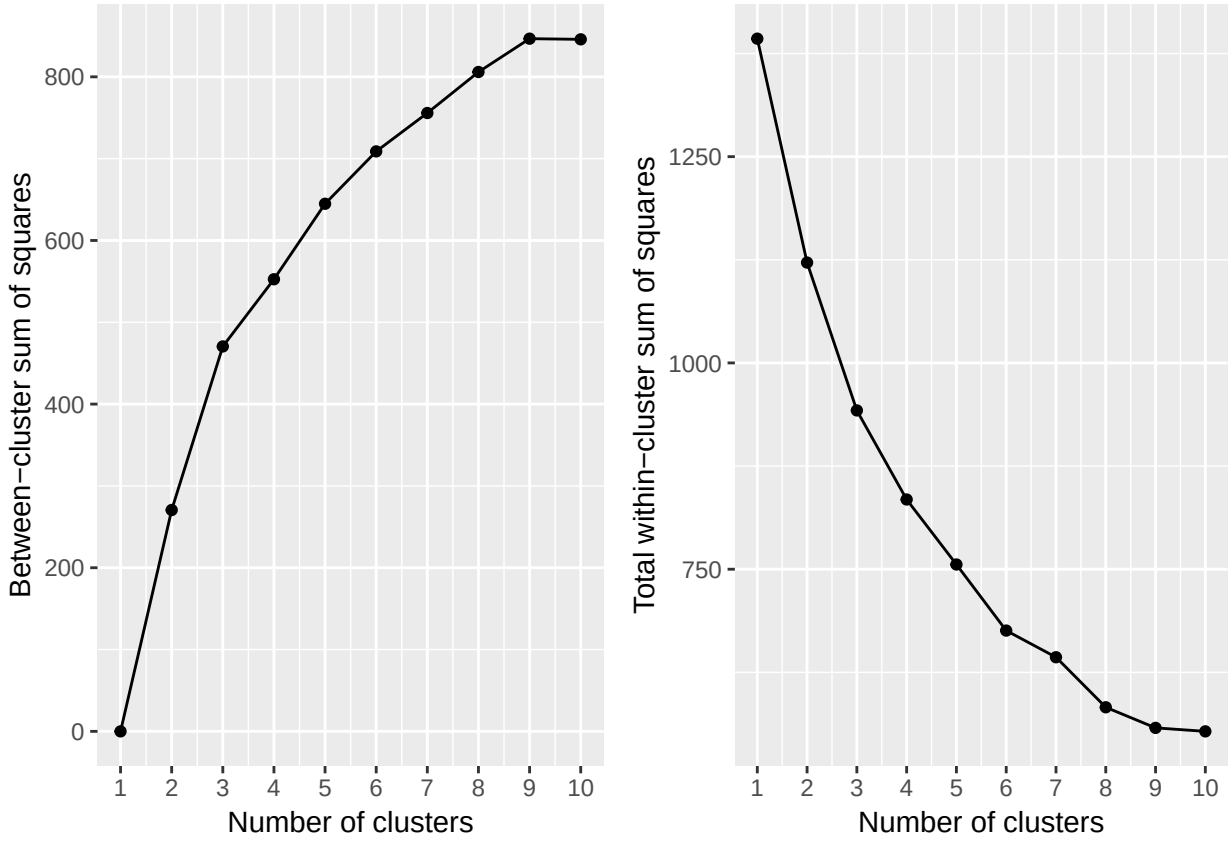
```
# Vendo os dados

ggplot(data.diabetes.rel.sel.new.norm,
               aes(x=num_medications + medical_specialty,
                   y=time_in_hospital + num_procedures + number_diagnoses + medical_specialty)) +
  geom_point() +
  labs(title="Dados normalizados")
```



Qual é o valor ideal para k? Deve-se escolher um número de clusters para que adicionar outro cluster não forneça uma partição muito melhor dos dados. Em algum momento, o ganho cairá, dando um ângulo no gráfico (critério do cotovelo). O número de clusters é escolhido neste momento. No nosso caso, é claro que 3 é o valor apropriado para k. Para estudar graficamente qual valor de k nos dá a melhor partição, podemos traçar entre e tot.withinss vs Choice de k.

```
bss <- numeric()
wss <- numeric()

# rodar o algoritmo com diferentes valores de K
set.seed(1234)

for(i in 1:10){

  # para casa k, calcula betweenss e tot.withinss
  bss[i] <- kmeans(data.diabetes.rel.sel.new.norm, centers=i)$betweenss
  wss[i] <- kmeans(data.diabetes.rel.sel.new.norm, centers=i)$tot.withinss
}

# Soma entre os quadrados dos quadrados vs Escolha de k

d3 <- qplot(1:10, bss, geom=c("point", "line"),
            xlab="Number of clusters", ylab="Between-cluster sum of squares") +
  scale_x_continuous(breaks=seq(0, 10, 1))

# Soma total de quadrados dentro do cluster vs Escolha de k
```

```
d4 <- qplot(1:10, wss, geom=c("point", "line"),
            xlab="Number of clusters", ylab="Total within-cluster sum of squares") +
  scale_x_continuous(breaks=seq(0, 10, 1))

# subplot
grid.arrange(d3, d4, ncol=2)
```



Execução do k-means

Com o algoritmo k-means identificamos a quantidade de grupos que meus dados formam e com isso podemos trazer semântica aos dados e tirar conclusões. De acordo com o método do cotovelo, a quantidade de clusters é igual 3, ou seja, o meu parâmetro **k**.

```
# selecionar somente os atributos de clusters
#data.diabetes.rel.sel.new.norm <- data.diabetes.rel.sel.new.norm[,c('time_in_hospital', 'num_medicatio

# Execução do K-means com k  = 6
set.seed(1234)
kmenas.diabetes <- kmeans(data.diabetes.rel.sel.new.norm, centers=3, nstart = 100)
```

*Vetor de inteiros indicando o custer ao qual cada ponto é alocado.

```
kmenas.diabetes$cluster
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   2   1   1   1   1   3   1   1   3   3   3   1   1   3   1   3   3   1
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
```

```
##   3   3   1   1   1   3   3   1   3   2   3   1   1   1   1   1   3   1
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   1   1   3   1   1   1   3   1   3   1   3   1   3   3   1   3   3   1
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   1   3   3   1   3   1   1   1   1   1   1   3   1   1   1   1   3   3
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   1   3   1   1   1   1   1   3   1   1   1   1   3   1   3   1   3   1
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   1   3   1   1   3   1   1   3   2   1   3   3   1   1   1   1   3   1
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   1   1   3   1   1   1   3   1   1   2   3   1   3   3   2   2   3   2
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   2   1   1   3   1   2   2   1   2   3   2   2   3   2   2   3   1   1
## 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
##   3   2   2   2   3   3   2   2   1   3   2   3   3   2   1   2   1   2
## 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   1   2   3   2   2   1   2   2   1   2   3   2   2   3   2   3   2   1
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
##   2   1   2   2   1   2   2   2   2   2   2   2   1   2   2   2   2   3
## 199 200
##   3   3
```

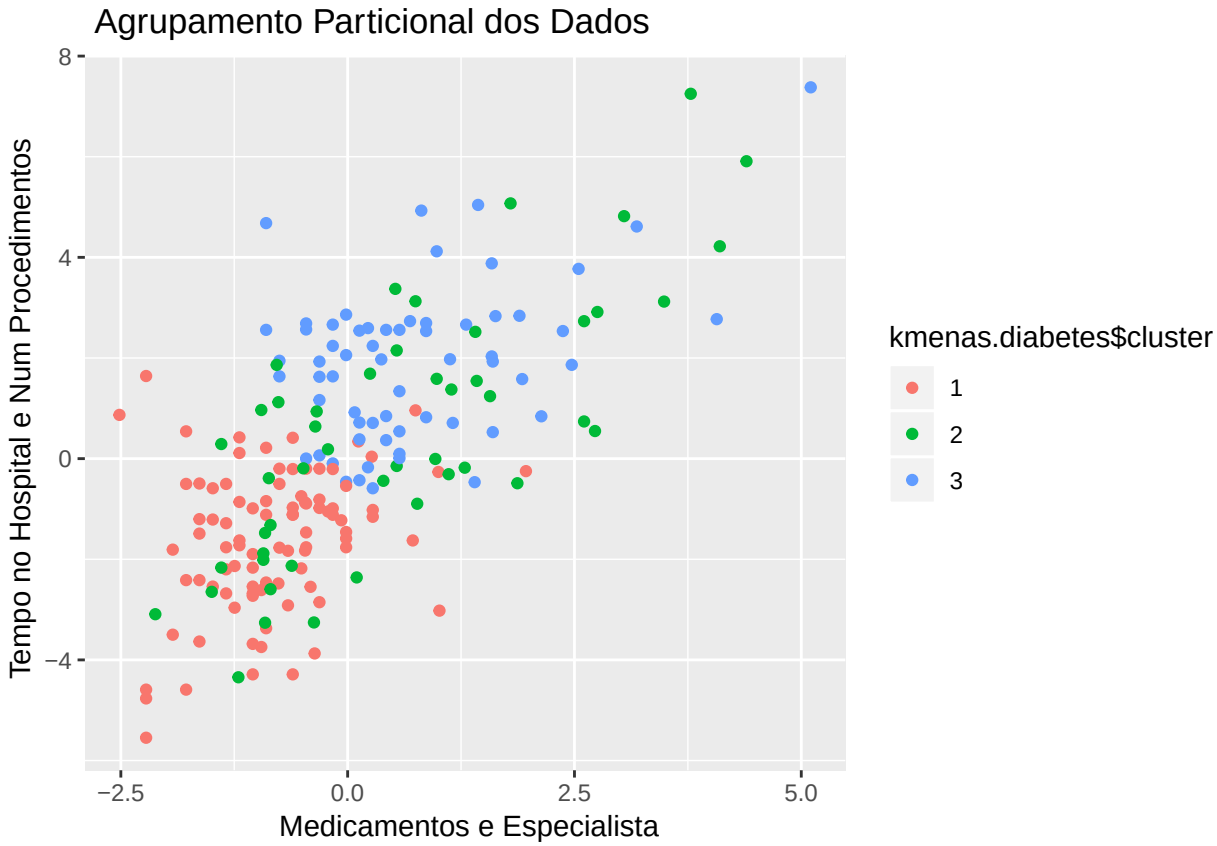
A matriz com o centro dos clusters.

```
kmenas.diabetes$centers
```

```
##   discharge_disposition_id admission_source_id time_in_hospital
## 1               -0.6769286          0.06165323      -0.53667929
## 2                1.4117950          0.02395162      -0.06237129
## 3               -0.1103642         -0.10803982       0.82733803
##   medical_specialty num_procedures num_medications number_diagnoses
## 1        -0.4590521     -0.4365631      -0.3538035       -0.1728929
## 2         1.1809940     -0.2813786      -0.5936599       -0.2884203
## 3        -0.2479519      0.8515621       0.9731933        0.4742667
```

Visualizando o agrupamento.

```
# melhorar os labels do meu grafico
kmenas.diabetes$cluster <- as.factor(kmenas.diabetes$cluster)

p1 <- ggplot(data.diabetes.rel.sel.new.norm,
             aes(
               x=num_medications + medical_specialty,
               y=time_in_hospital + num_procedures + number_diagnoses + medical_specialty,
               color = kmenas.diabetes$cluster)) +
      geom_point() +
      xlab(" Medicamentos e Especialista") +
      ylab(" Tempo no Hospital e Num Procedimentos") +
      labs(title=" Agrupamento Particional dos Dados ")

p1
```



Redução de Dimensionalidade

O conjunto de dados, mesmo com a seleção manual dos atributos e visualização, ainda se tem como hipótesi alguns atributos que juntos podem inferir melhores resultados. Para diminuir o custo e aumentar a precisão do cluster, uma alternativa é aplica a redução de dimensionalidade. Uma técnica muito comum utilizada é o Principal Component Analysis.

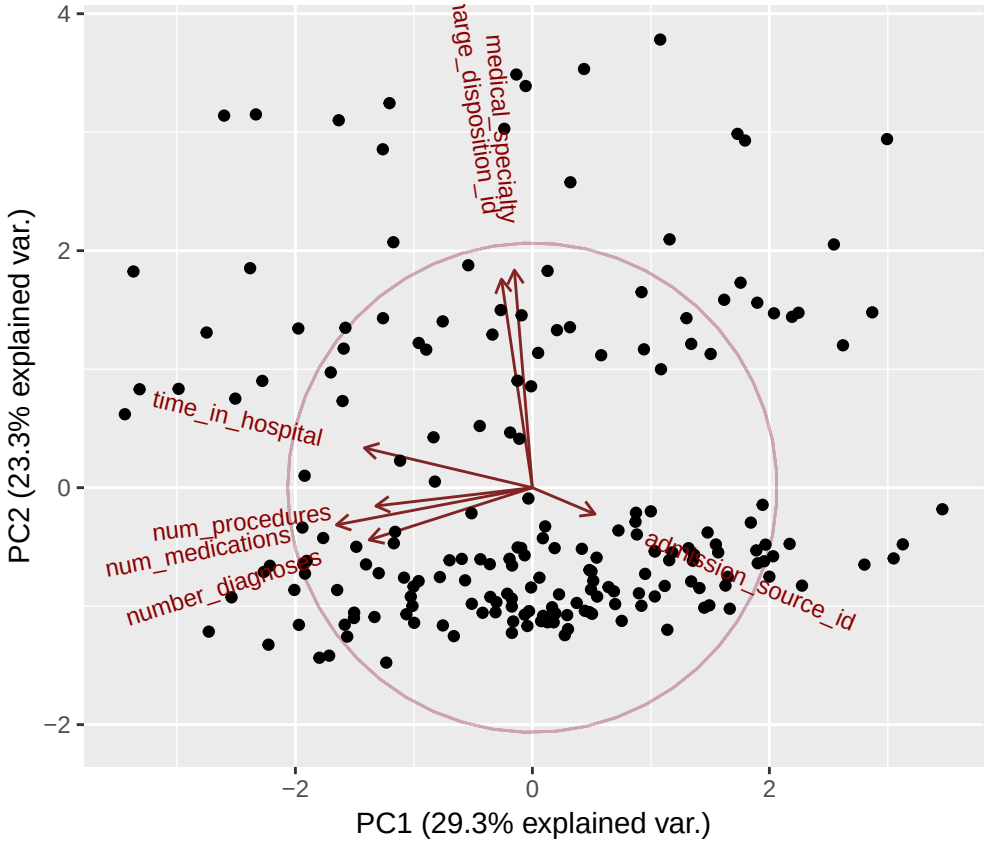
Principal Component Analysis - PCA

- Método que utiliza uma transformação ortogonal para converter um conjunto de observações correlacionadas em um conjunto de componentes principais (observações linearmente não-correlacionadas). São formadas combinações lineares das variáveis observadas e gerados componentes componentes na mesma quantidade de atributos, sendo que:
  - O primeiro componente principal consiste na combinação que responde pela maior quantidade de variância na amostra.
  - O segundo componente responde pela segunda maior variância na amostra e não é correlacionado com o primeiro componente.
  - Sucessivos componentes explicam progressivamente menores porções de variância total da amostra e todos são não correlacionados uns aos outros.

```
# aplica o PCA - scale = TRUE é aconselhavel
# mas o padrão é false
data.diabetes.rel.sel.new.norm.pca <- prcomp(data.diabetes.rel.sel.new.norm, center = TRUE, scale. = TRU

data.diabetes.rel.sel.new.norm.pca
```

```
## Standard deviations:
## [1] 1.4333023 1.2767350 1.0851562 0.8484437 0.7647333 0.6655109 0.6248599
##
## Rotation:
##                                PC1         PC2          PC3         PC4
## discharge_disposition_id -0.08814164  0.66775872 -0.070944175  0.31144236
## admission_source_id       0.17866472 -0.08463866 -0.777308084  0.29099297
## time_in_hospital         -0.47910264  0.12722030 -0.340765111 -0.55757826
## medical_specialty        -0.05129945  0.69650049  0.003974293 -0.02560464
## num_procedures           -0.44632072 -0.05870094  0.463859660  0.27714826
## num_medications          -0.55888622 -0.11917025 -0.172346447 -0.21338542
## number_diagnoses         -0.46538480 -0.16730007 -0.172491533  0.62002672
##                                PC5        PC6         PC7
## discharge_disposition_id -0.14279432  0.2763677 -0.58951724
## admission_source_id      -0.46907696 -0.1583595  0.16407166
## time_in_hospital          0.07840128 -0.4891320 -0.28611954
## medical_specialty         0.11839733 -0.1288725  0.69351135
## num_procedures           -0.61066849 -0.3568663  0.07145270
## num_medications          -0.19506085  0.7100247  0.23669434
## number_diagnoses          0.57310310 -0.1058123  0.03997271
```

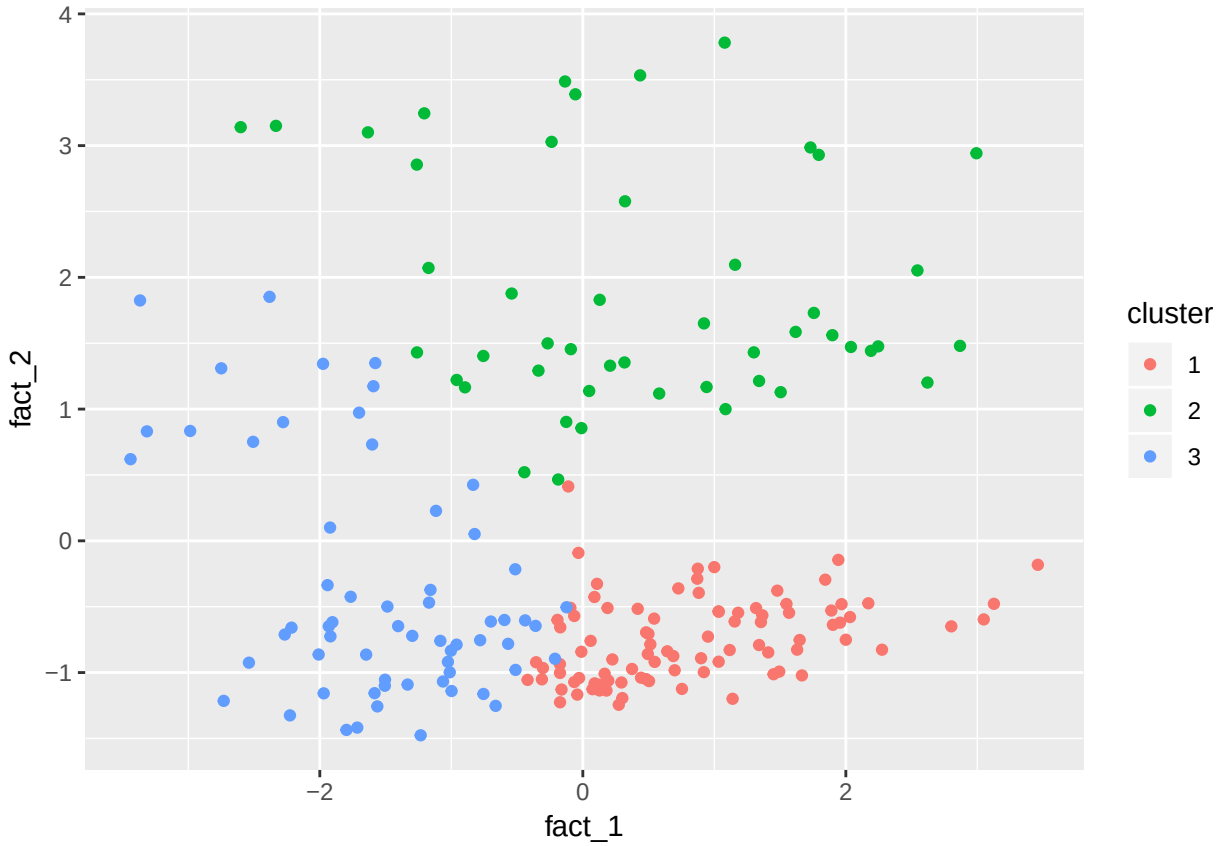
```
library(ggbiplot)
ggbiplot(data.diabetes.rel.sel.new.norm.pca, obs.scale = 1, var.scale = 1,
   ellipse = TRUE, circle = TRUE) +
  scale_color_discrete(name = '') +
  theme(legend.direction = 'horizontal', legend.position = 'top')
```



Os dois primeiros componentes descrevem 52,6% da variancia da amostra.

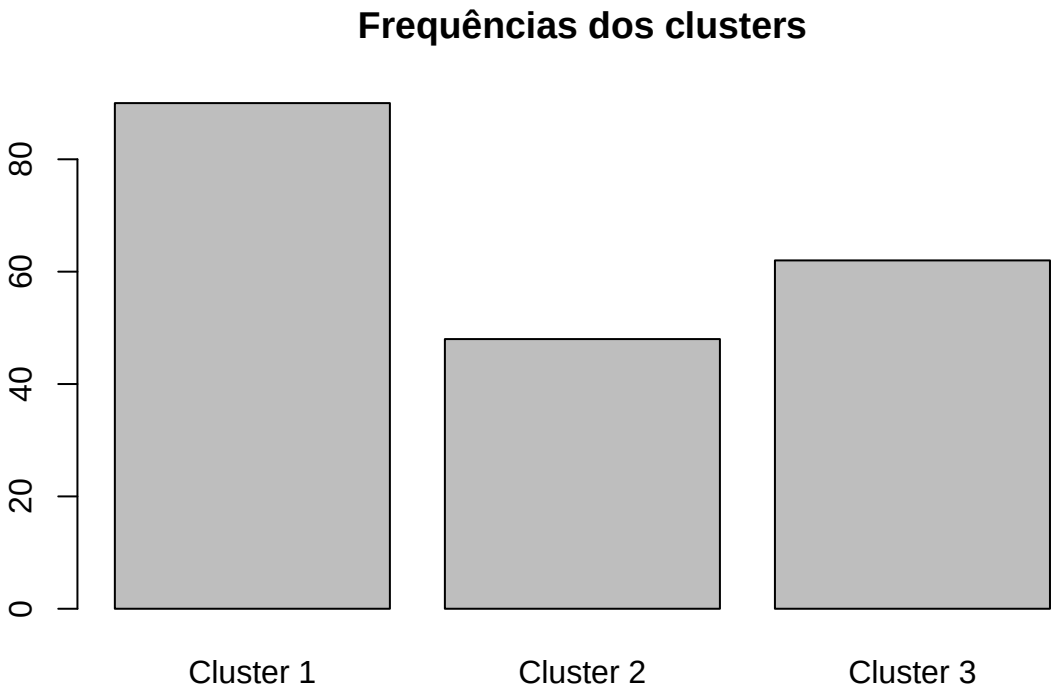
Visualizado os grupos formados com o meu cluster.

```
library(ggplot2)
# colletando os dados
tmp_d = data.frame(matrix(ncol=0, nrow=nrow(data.diabetes.rel.sel.new.norm)))
tmp_d$cluster = as.factor(kmenas.diabetes$cluster)
tmp_d$fact_1 = as.numeric(data.diabetes.rel.sel.new.norm.pca$x[, 1])
tmp_d$fact_2 = as.numeric(data.diabetes.rel.sel.new.norm.pca$x[, 2])
tmp_d$label = rownames(data.diabetes.rel.sel.new.norm)

# visualizando o agrupamento depois da reduçao de dimensionalidade
ggplot(tmp_d, aes(fact_1, fact_2, color = cluster)) +
    geom_point()
```



Visulizando a quantidade de elementos que cada centroide agrupou.

```
kmenas.diabetes$cluster%>%
  table()%>%
  barplot(main="Frequências dos clusters", names.arg=c("Cluster 1", "Cluster 2", "Cluster 3"))
```



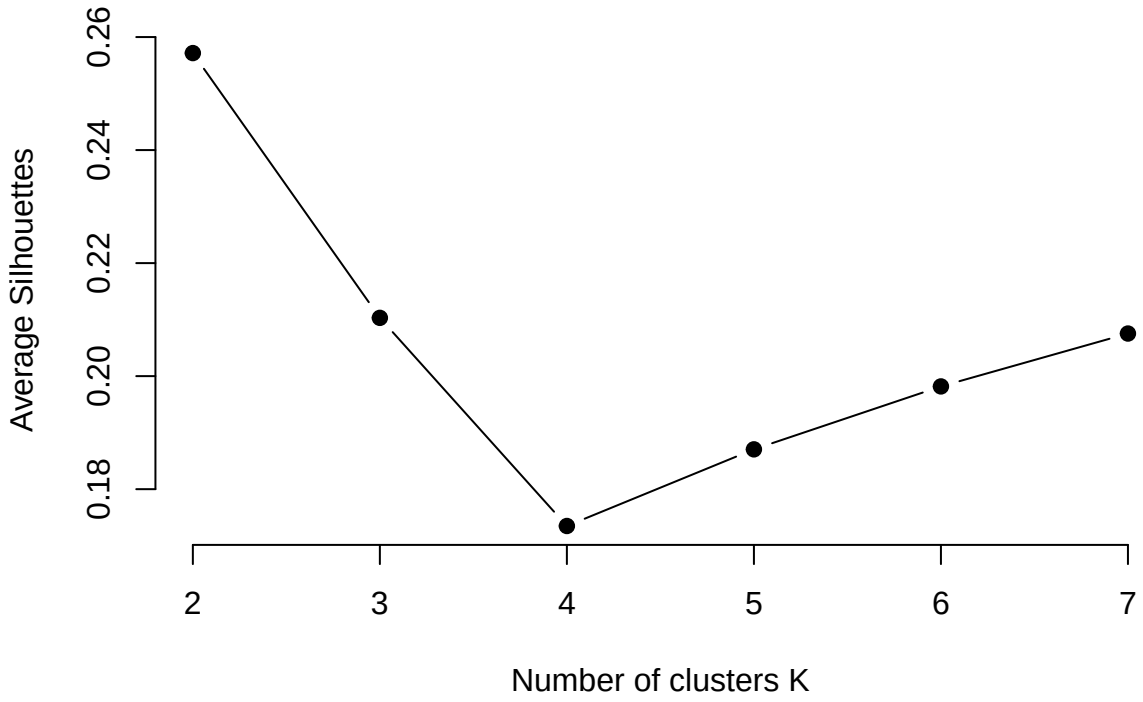
Validação

Aqui valida-se o quão o método conseguio agrupar conforme um índice de validação. Validar com critérios internos, pois vai medir a qualidade do agrupamento com base nos dados originais, ja que, os dados não posuem rótulos ou estruturas definidas.

- **Silhueta**
  - Baseia-se na similaridade entre objetos do mesmo grupo e na distância entre objetos de um cluster e objetos do cluster mais próximo;
  - Quando a silhueta é calculada para cada objeto, seu valor será próximo de 1, se o objeto está bem situado dentro do seu cluster;
  - Valor próximo de -1 indica que o objeto deveria estar em outro cluster;

```
library(purrr)
library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms

# function to compute average silhouette for k clusters
avg_sil <- function(k) {
  km.res <- kmeans(data.diabetes.rel.sel.new.norm, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data.diabetes.rel.sel.new.norm))
  mean(ss[, 3])
}

# Compute and plot wss for k = 2 to k = 15
k.values <- 2:7

# extract avg silhouette for 2-15 clusters
avg_sil_values <- map_dbl(k.values, avg_sil)

plot(k.values, avg_sil_values,
```

```
       type = "b", pch = 19, frame = FALSE,
       xlab = "Number of clusters K",
       ylab = "Average Silhouettes")
```



O Gráfico acima calcula a silheura para clusters de 1-7. Os resultados mostram que 2 clusters maximizam os valores médios de silhueta com 3 clusters entrando como o segundo número ideal de clusters.

Agrupamento com SOM

Antes de criarmos um SOM, precisamos escolher em quais variáveis queremos pesquisar padrões.

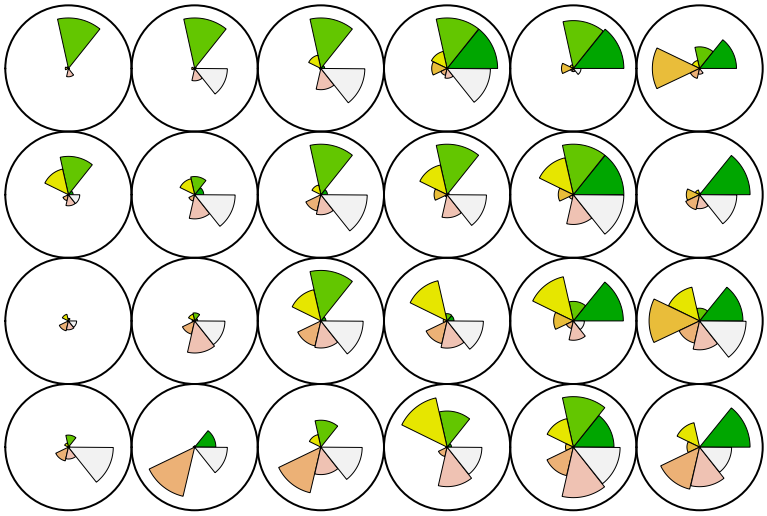
```
require(kohonen)
```

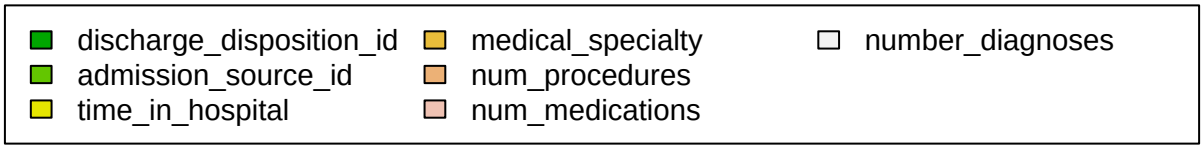
```
## Loading required package: kohonen
## Loading required package: class
## Loading required package: MASS
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##     select
##
## Attaching package: 'kohonen'
## The following object is masked from 'package:purrr':
##
##     map
```

```
require(RColorBrewer)
```

```
## Loading required package: RColorBrewer
```

```
diabetes.SOM1 <- som(scale(data.diabetes.rel.sel.new.norm), grid = somgrid(6, 4, "rectangular"))
plot(diabetes.SOM1)
```



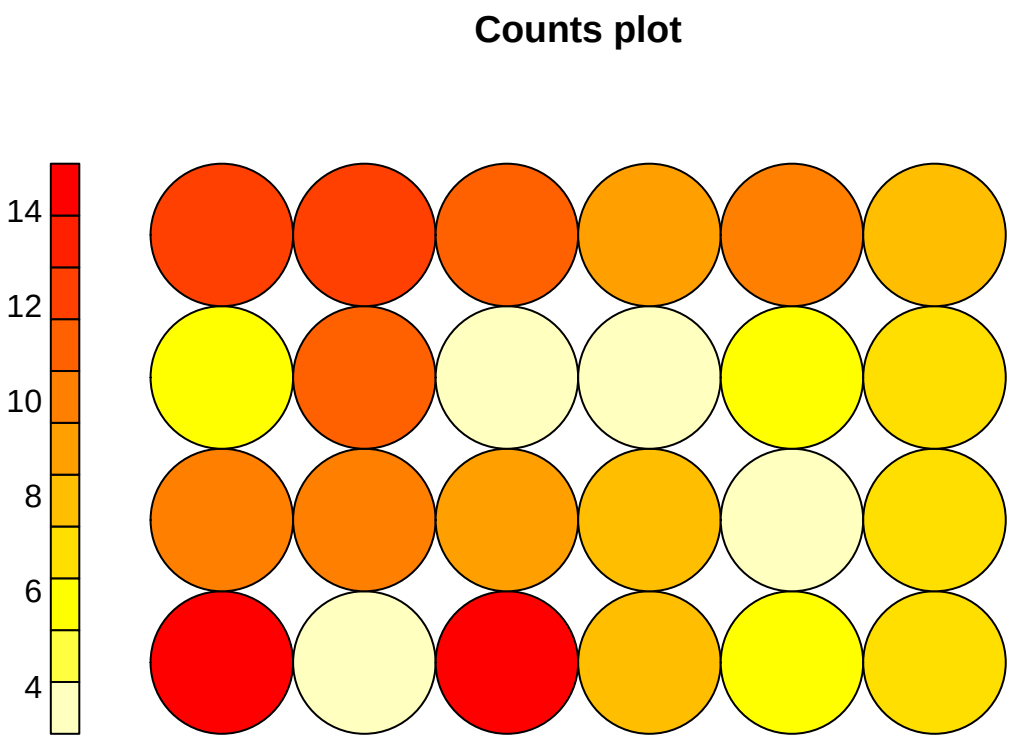


Observe que escalamos e centralizamos nossos dados de treinamento e definimos o tamanho e a organização da grade. O gráfico padrão Kohonen SOM cria essas representações de torta dos vetores representativos para as células da grade, onde o raio de uma cunha corresponde à magnitude em uma determinada dimensão.

Heatmap SOM

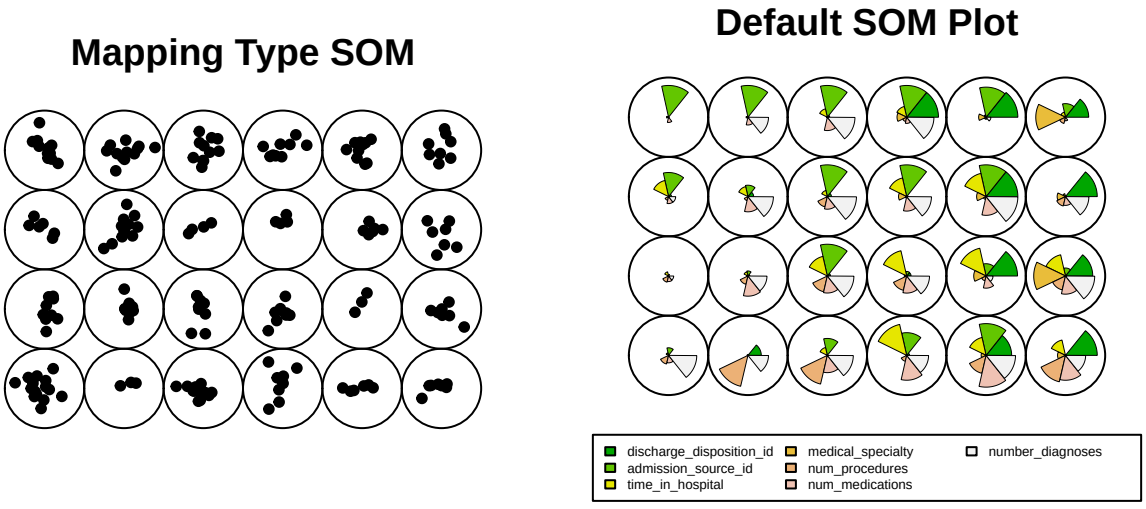
Lembre-se de que o acima é apenas um mapa dos atributos - cada célula exibe seu vetor representativo. O tipo de contagem do SOM faz exatamente isso e cria um mapa de calor baseado no número de atribuídos a cada célula.

```
# reverse color ramp
colors <- function(n, alpha = 1) {
    rev(heat.colors(n, alpha))
}

plot(diabetes.SOM1, type = "counts", palette.name = colors, heatkey = TRUE)
```



Alternativamente, pode-se plotar os atributos como pontos na grade usando o tipo de mapeamento SOM.

```
par(mfrow = c(1, 2))
plot(diabetes.SOM1, type = "mapping", pchs = 20, main = "Mapping Type SOM")
plot(diabetes.SOM1, main = "Default SOM Plot")
```



Conclusão

- Os grupos podem indicar a variedade do estado de saúde das pessoas com diabétes, ou seja, com diferentes graus, leve, moderado, normal, grave e diabete melitus.
- Os vastos grupos indicam que o tratamento merece mais cuidados.
- Pode-se concluir que diante das diversas características dos paciente, a readmissão dos pacientes acontecem nos mais diversos casos da diabete, é uma doença severa e que merece uma atenção e

tratamento adequado, sendo grande parte resposável o própio paciente a seus limites.

Referências

- Introdução ao Aprendizado de Máquina
- Origem do Dataset
- Descrição dos Atributos
- Silhueta
- Silhueta e Cotovelo
- Indroduction Data Mining
- Chapter 8
- SOM