Problema

Existe uma hipótese de que o referenciamento de pacientes ao HC não é totalmente necessário.
casos onde o paciente poderia ser tratado na UBS, pois o HC é um hospital de alta complexidad

Hipótese desta análise(opcional)

Esta analise tem uma hipotese de que a efetividade do tratamento de uma equipe esta correlaci
ao fato de ela ter um protocolo efetivo

Importando bibliotecas principais

```
import numpy  as np
import pandas as pd
import matplotlib.pyplot as pl
import seaborn as sb
import random, decimal

%matplotlib inline
```

```
pip install bokeh
```

```
Requirement already satisfied: bokeh in /usr/local/lib/python3.6/dist-packages (1.0.4
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: numpy>=1.7.1 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: packaging>=16.8 in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: tornado>=4.3 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: pillow>=4.0 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: six>=1.5.2 in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: Jinja2>=2.7 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.6/dist-pack
Requirement already satisfied: olefile in /usr/local/lib/python3.6/dist-packages (fro
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6/dist-pack
```

```
from bokeh.io import output_notebook
output_notebook()
```

Importando e Explorando o dataset

```
df = pd.read_csv('dsAnamneseFechada.csv', parse_dates=['DAT_HORA_ATENDIMENTO','DAT_HORA_PR
```

```
df.dtypes
```

```
DAT_HORA_ATENDIMENTO          datetime64[ns]
NOM_ENCAMINHAMENTO                    object
NOM_MODALIDADE_ATENDIMENTO            object
NOM_MUNICIPIO                         object
NOM_EQUIPE                            object
NOM_TIPO_CASO                         object
IDADE                                float64
COD_CID                               object
DAT_HORA_PREVISTA                     object
DAT_HORA_EVOLUCAO                     object
DAT_HORA_ANAMNESE             datetime64[ns]
DAT_HORA_ALTA                         object
QTD_EVOLUCAO                           int64
DAT_ULTIMA_EVOLUCAO           datetime64[ns]
dtype: object
```

verificando escopos da modalidade

```
df["NOM_MODALIDADE_ATENDIMENTO"].value_counts()
```

```
AMBULATORIO         47634
INTERNAÇÃO           1070
SADT EXTERNO          345
SADT UBS MARILIA      138
Name: NOM_MODALIDADE_ATENDIMENTO, dtype: int64
```

escopos de equipe

```
df["NOM_EQUIPE"].value_counts()
```

```
AMBULATÓRIO SAÚDE MENTAL             11266
ORTOPEDIA E TRAUMATOLOGIA             4210
OFTALMOLOGIA                          4049
ENDOCRINOLOGIA E METABOLISMO          3404
NEUROLOGIA                            2410
CIRURGIA VASCULAR                     2374
ONCOLOGIA CLÍNICA                     2268
DERMATOLOGIA                          2031
REUMATOLOGIA                          1700
ONCO-HEMATOLOGIA INFANTIL             1667
OTORRINOLARINGOLOGIA                  1572
UROLOGIA                              1078
HEMATOLOGIA ADULTO                    1053
GINECOLOGIA GERAL                      983
CARDIOLOGIA                            962
PNEUMOLOGIA                            831
AMB PEDIATRIA ESPECIALIZADA            818
CIRURGIA GERAL E DO TRAUMA             767
CIRURGIA PLÁSTICA                      740
OBSTETRÍCIA                            662
GASTROENTEROLOGIA - CLÍNICA MÉDICA     521
INFECTOLOGIA                           505
NEFROLOGIA                             504
SERVIÇO DE APOIO AO COLABORADOR        425
NEUROCIRURGIA                          360
GERIATRIA                              346
GASTROENTEROLOGIA CIRÚRGICA            298
CIRURGIA CABEÇA E PESCOÇO              283
CENTRO DE INFUSÃO                      193
ONCO GINECOLOGIA                       179
RADIOTERAPIA                           169
CIRURGIA CARDÍACA                      133
CIRURGIA TORÁCICA                      131
QUIMIOTERAPIA ADULTO                   102
MEDICINA INTERNA                        38
SERVIÇO DE NUTRIÇÃO E DIETÉTICA         35
GENÉTICA                                25
UROLÓGIA                                25
IMUNOPATOLOGIA CLÍNICA E ALÉRGICA       20
ONCOCLÍNICA                             18
PRÉ-OPERATÓRIO                          13
PSICOLOGIA HOSPITALAR                    6
CLÍNICA MÉDICA ESPECIALIZADA             5
ENFERMAGEM                               3
HEMOTERAPIA                              2
CENTRO CIRÚRGICO                         2
BRONCOSCOPIA                             1
Name: NOM_EQUIPE, dtype: int64
```

verificando escopos dos dias da semana (0=segunda,1=terça,etc..)

```
df['DIASEMANA'] = df['DAT_HORA_ATENDIMENTO'].dt.dayofweek

df["DIASEMANA"].value_counts()
```

```
0    10180
2     9698
3     9032
1     8690
4     6844
5     2474
6     2269
Name: DIASEMANA, dtype: int64
```

```
df['DAT_HORA_ATENDIMENTO'].describe()
```

```
count                   49187
unique                  11560
top       2018-06-28 07:00:00
freq                      108
first     2018-01-02 07:00:00
last      2018-12-28 12:10:00
Name: DAT_HORA_ATENDIMENTO, dtype: object
```

Limpeza e Tratamento de dados

```
#utilizando dados somente de 2018
df2018 = df[(df['DAT_HORA_ATENDIMENTO'] > '2018-1-1') & (df['DAT_HORA_ATENDIMENTO'] <= '20
```

```
#filtrando somente as equipes com maior incidencia
dfLimpo = df2018[df2018['NOM_EQUIPE'].map(df2018['NOM_EQUIPE'].value_counts()) > 2000]

#tirar os SESMT e SASCe saude mental
dfLimpo = dfLimpo[dfLimpo.NOM_EQUIPE!='AMBULATÓRIO SAÚDE MENTAL']
```

```
dfLimpo["NOM_MODALIDADE_ATENDIMENTO"].value_counts()
```

```
AMBULATORIO         20146
INTERNAÇÃO            389
SADT EXTERNO          155
SADT UBS MARILIA       56
Name: NOM_MODALIDADE_ATENDIMENTO, dtype: int64
```

```
#atribuir o valor de protocolo efetivo para a ENDOCRINO
import random

def getProtocolo(equipe):
  if (equipe=='ENDOCRINOLOGIA E METABOLISMO'):
    return 1 + (random.randint(0, 200)/1000)
  elif (equipe=='REUMATOLOGIA'):
    return 0.5  + (random.randint(0, 200)/1000)
  else:
    return 0 + (random.randint(0, 200)/1000)

dfLimpo['PROTOCOLO'] = dfLimpo.apply(lambda row: getProtocolo(row.NOM_EQUIPE), axis = 1)
```

```
dfLimpo['PROTOCOLO'] = dfLimpo.apply(lambda row: getProtocolo(row.NOM_EQUIPE), axis = 1)
dfLimpo['DURACAO'] = dfLimpo['DAT_ULTIMA_EVOLUCAO'].sub(dfLimpo['DAT_HORA_ANAMNESE'], axis
dfLimpo['NDURACAO'] = dfLimpo['DURACAO'] / np.timedelta64(1, 'D')

dfLimpo['NDURACAO'] = dfLimpo['DURACAO'] / np.timedelta64(1, 'D')

dfLimpo = dfLimpo[dfLimpo.NOM_EQUIPE!='AMBULATÓRIO SAÚDE MENTAL']

dfFiltro = dfLimpo[dfLimpo.NOM_MODALIDADE_ATENDIMENTO=='SADT EXTERNO']

dfFiltro
```

	DAT_HORA_ATENDIMENTO	NOM_ENCAMINHAMENTO	NOM_MODALIDADE_ATENDIMENTO	NOM_MUN
123	2018-04-18 12:01:00	RETORNO	SADT EXTERNO	A
167	2018-09-08 12:00:00	RETORNO	SADT EXTERNO	BR
212	2018-08-06 12:02:00	RETORNO	SADT EXTERNO	MORT
829	2018-11-13 09:00:00	ALTA	SADT EXTERNO	M
1292	2018-08-27 07:00:00	RETORNO	SADT EXTERNO	M
...	...	...	...	
47126	2018-03-26 07:01:00	RETORNO	SADT EXTERNO	
47359	2018-08-10 07:05:00	RETORNO	SADT EXTERNO	MONTE DE
47572	2018-03-07 07:00:00	RETORNO	SADT EXTERNO	
47825	2018-06-21 07:05:00	RETORNO	SADT EXTERNO	M
48323	2018-10-07 07:00:00	RETORNO	SADT EXTERNO	L

155 rows × 18 columns

```
dfFiltro
```

	DAT_HORA_ATENDIMENTO	NOM_ENCAMINHAMENTO	NOM_MODALIDADE_ATENDIMENTO	NOM_MUN
123	2018-04-18 12:01:00	RETORNO	SADT EXTERNO	A
167	2018-09-08 12:00:00	RETORNO	SADT EXTERNO	BR
212	2018-08-06 12:02:00	RETORNO	SADT EXTERNO	MORT
829	2018-11-13 09:00:00	ALTA	SADT EXTERNO	M
1292	2018-08-27 07:00:00	RETORNO	SADT EXTERNO	M
...	...	...	...	
47126	2018-03-26 07:01:00	RETORNO	SADT EXTERNO	
47359	2018-08-10 07:05:00	RETORNO	SADT EXTERNO	MONTE DE
47572	2018-03-07 07:00:00	RETORNO	SADT EXTERNO	
47825	2018-06-21 07:05:00	RETORNO	SADT EXTERNO	M
48323	2018-10-07 07:00:00	RETORNO	SADT EXTERNO	L

155 rows × 18 columns

Profiling

```
import pandas_profiling as pp
pp.ProfileReport(dfLimpo)
```

```
/usr/local/lib/python3.6/dist-packages/pandas_profiling/describe.py:392: FutureWarnir
  variable_stats = pd.concat(ldesc, join_axes=pd.Index([names]), axis=1)
```

Overview

Dataset info

Number of variables	19
Number of observations	20746
Total Missing (%)	9.3%
Total size in memory	3.0 MiB
Average record size in memory	152.0 B

Variables types

Numeric	6
Categorical	10
Boolean	0
Date	3
Text (Unique)	0
Rejected	0
Unsupported	0

Warnings

- NOM_MUNICIPIO has a high cardinality: 1110 distinct values Warning
- COD_CID has a high cardinality: 1150 distinct values Warning
- DAT_HORA_PREVISTA has 16614 / 80.1% missing values Missing
- DAT_HORA_PREVISTA has a high cardinality: 1553 distinct values Warning
- DAT_HORA_EVOLUCAO has a high cardinality: 12658 distinct values Warning
- DAT_HORA_ALTA has 20092 / 96.8% missing values Missing
- DAT_HORA_ALTA has a high cardinality: 520 distinct values Warning
- DIASEMANA has 4928 / 23.8% zeros Zeros
- DURACAO has a high cardinality: 10661 distinct values Warning

Variables

index

Numeric

Distinct count	20746
Unique (%)	100.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	24565
Minimum	3

Maximum	49183
Zeros (%)	0.0%

Toggle details

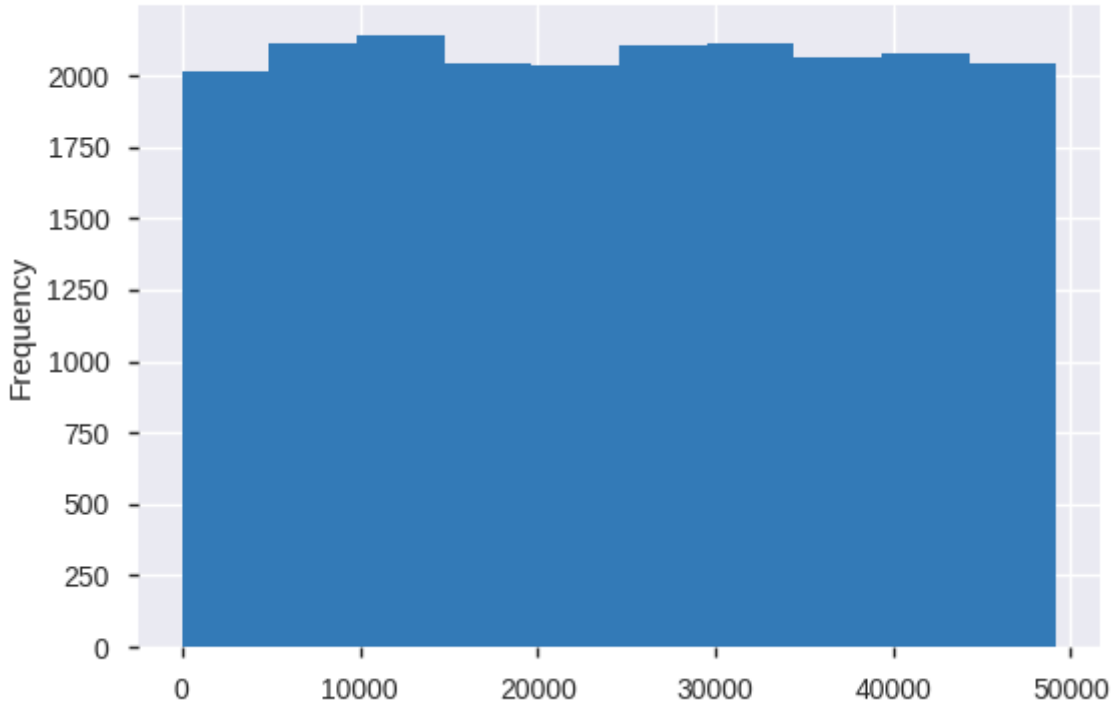
- Statistics
- Histogram
- Common Values
- Extreme Values

Quantile statistics

Minimum	3
5-th percentile	2637.8
Q1	12220
Median	24698
Q3	36810
95-th percentile	46686
Maximum	49183
Range	49180
Interquartile range	24590

Descriptive statistics

Standard deviation	14168
Coef of variation	0.57677
Kurtosis	-1.2009
Mean	24565
MAD	12278
Skewness	0.0019718
Sum	509629081
Variance	200740000
Memory size	162.2 KiB



Value	Count	Frequency (%)
34815	1	0.0%
13043	1	0.0%
33493	1	0.0%
37591	1	0.0%
25305	1	0.0%
31450	1	0.0%
29403	1	0.0%
23262	1	0.0%
38212	1	0.0%
10976	1	0.0%
Other values (20736)	20736	100.0%

Minimum 5 values

Value	Count	Frequency (%)
3	1	0.0%
7	1	0.0%
8	1	0.0%
10	1	0.0%
14	1	0.0%

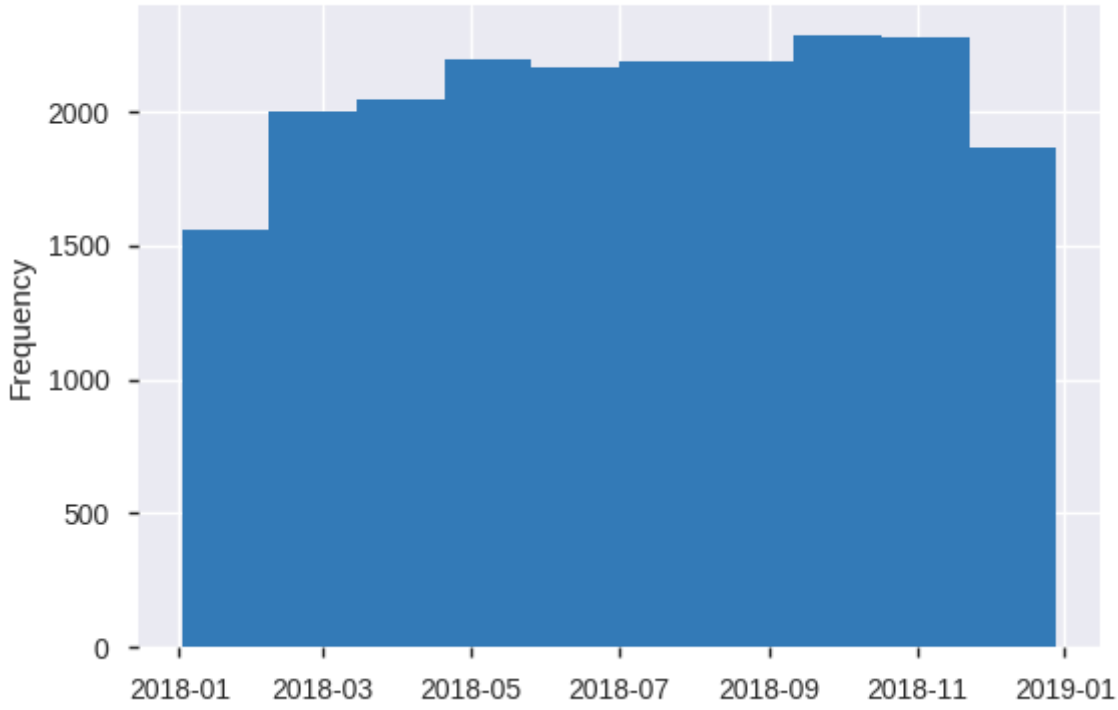
Maximum 5 values

Value	Count	Frequency (%)
49172	1	0.0%
49178	1	0.0%
49179	1	0.0%
49182	1	0.0%
49183	1	0.0%

DAT_HORA_ATENDIMENTO
Date

Distinct count	6640
Unique (%)	32.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Minimum	2018-01-02 07:00:00
Maximum	2018-12-28 12:10:00

Toggle details



NOM_ENCAMINHAMENTO
Categorical

Distinct count	20
Unique (%)	0.1%
Missing (%)	0.0%
Missing (n)	0

RETORNO	19781
ALTA	684
AGUARDANDO CIRURGIA	80
Other values (17)	201

Toggle details

Value	Count	Frequency (%)
RETORNO	19781	95.3%
ALTA	684	3.3%
AGUARDANDO CIRURGIA	80	0.4%
PEDIDO DE INTERNAÇÃO HC-I	69	0.3%
RETORNO E ENCAMINHAMENTO	41	0.2%
ALTA E ENCAMINHAMENTO	38	0.2%
FALTA A CONSULTA AGENDADA	22	0.1%
CONTRA-REFERENCIA	10	0.0%
ENCAM.UBS/PSF DE ORIGEM	4	0.0%
URG./EMERG. HCI	4	0.0%
Other values (10)	13	0.1%

NOM_MODALIDADE_ATENDIMENTO
Categorical

Distinct count	4
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0

AMBULATORIO		20146
INTERNAÇÃO	389	
SADT EXTERNO	155	

Toggle details

Value	Count	Frequency (%)
AMBULATORIO	20146	97.1%
INTERNAÇÃO	389	1.9%
SADT EXTERNO	155	0.7%
SADT UBS MARILIA	56	0.3%

NOM_MUNICIPIO
Categorical

Distinct count	1110
Unique (%)	5.4%
Missing (%)	0.0%
Missing (n)	0

MARILIA		5474
GARÇA	912	
TUPÃ	660	
Other values (1107)		13700

Toggle details

Value	Count	Frequency (%)
MARILIA	5474	26.4%
GARÇA	912	4.4%
TUPÃ	660	3.2%
POMPÉIA	586	2.8%
VERA CRUZ	579	2.8%
SÃO PAULO	394	1.9%
ORIENTE	379	1.8%
GÁLIA	360	1.7%
ASSIS	340	1.6%

ADAMANTINA	315	1.5%
Other values (1100)	10747	51.8%

NOM_EQUIPE
Categorical

Distinct count	7
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0

ORTOPEDIA E TRAUMATOLOGIA	4210
OFTALMOLOGIA	4049
ENDOCRINOLOGIA E METABOLISMO	3404
Other values (4)	9083

Toggle details

Value	Count	Frequency (%)
ORTOPEDIA E TRAUMATOLOGIA	4210	20.3%
OFTALMOLOGIA	4049	19.5%
ENDOCRINOLOGIA E METABOLISMO	3404	16.4%
NEUROLOGIA	2410	11.6%
CIRURGIA VASCULAR	2374	11.4%
ONCOLOGIA CLÍNICA	2268	10.9%
DERMATOLOGIA	2031	9.8%

NOM_TIPO_CASO
Categorical

Distinct count	34
Unique (%)	0.2%
Missing (%)	0.0%
Missing (n)	0

RETORNO	11189
AGENDADO PELO PROFISSIONAL	2686
AGENDADO	2541
Other values (31)	4330

Toggle details

Value	Count	Frequency (%)

RETORNO	11189	53.9%
AGENDADO PELO PROFISSIONAL	2686	12.9%
AGENDADO	2541	12.2%
QUIMIOTERAPIA	983	4.7%
ENCAIXE AUTORIZADO	788	3.8%
RETORNO MÉDICO	677	3.3%
RETORNO FALTOSOS	386	1.9%
REGULAÇÃO INTERNA	261	1.3%
NOVO	235	1.1%
SUS	219	1.1%
Other values (24)	781	3.8%

IDADE

Numeric

Distinct count	8569
Unique (%)	41.3%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	55.558
Minimum	0.60153
Maximum	100.94
Zeros (%)	0.0%

Toggle details

- Statistics
- Histogram
- Common Values
- Extreme Values

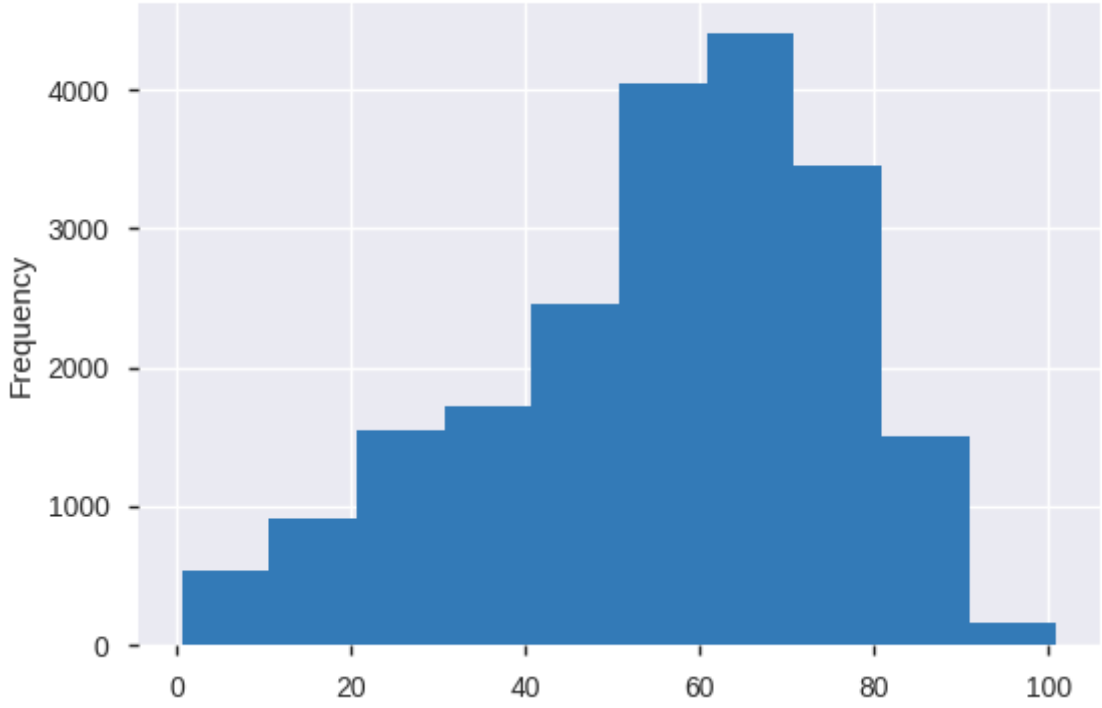
Quantile statistics

Minimum	0.60153
5-th percentile	16.979
Q1	43.086
Median	58.947
Q3	70.68
95-th percentile	83.876
Maximum	100.94
Range	100.34
Interquartile range	27.594

Descriptive statistics

Standard deviation	20.264

Coef of variation	0.36473
Kurtosis	-0.28285
Mean	55.558
MAD	16.344
Skewness	-0.57992
Sum	1152600
Variance	410.62
Memory size	162.2 KiB



Value	Count	Frequency (%)
58.946735984271896	108	0.5%
90.15495516235409	98	0.5%
70.9165989979706	46	0.2%
56.0426263952308	30	0.1%
82.5604346144089	28	0.1%
29.270023655504797	28	0.1%
50.2207085870117	24	0.1%
45.322078450025394	24	0.1%
52.075503107559605	22	0.1%
63.659064751395206	21	0.1%
Other values (8559)	20317	97.9%

Minimum 5 values

Value	Count	Frequency (%)
0.6015305048198879	5	0.0%
0.933037354134957	1	0.0%
1.00427023084729	1	0.0%
1.01796886098427	1	0.0%
1.04810584728564	1	0.0%

Maximum 5 values

Maximum 5 values

Value	Count	Frequency (%)
99.9111195459158	2	0.0%
99.99331132673771	2	0.0%
100.083722285642	2	0.0%
100.563174340436	2	0.0%
100.93851680619	3	0.0%

COD_CID
Categorical

Distinct count	1150
Unique (%)	5.5%
Missing (%)	0.0%
Missing (n)	0

Z988	2045
Z010	1290
L989	786
Other values (1147)	16625

Toggle details

Value	Count	Frequency (%)
Z988	2045	9.9%
Z010	1290	6.2%
L989	786	3.8%
E119	739	3.6%
Z000	673	3.2%
H409	615	3.0%
C509	464	2.2%
E039	445	2.1%
I702	434	2.1%
E109	428	2.1%
Other values (1140)	12827	61.8%

DAT_HORA_PREVISTA
Categorical

Distinct count	1553
Unique (%)	7.5%
Missing (%)	80.1%
Missing (n)	16614

23/04/2019 07:00:00	41

11/03/2019 07:00:00	25
22/01/2019	21
Other values (1549)	4045
(Missing)	16614

Toggle details

Value	Count	Frequency (%)
23/04/2019 07:00:00	41	0.2%
11/03/2019 07:00:00	25	0.1%
22/01/2019	21	0.1%
09/01/2019 07:00:00	20	0.1%
25/03/2019	19	0.1%
03/04/2019 07:00:00	19	0.1%
15/04/2019 07:00:00	18	0.1%
18/03/2019	18	0.1%
23/04/2019 12:00:00	18	0.1%
23/01/2019 07:00:00	18	0.1%
Other values (1542)	3915	18.9%
(Missing)	16614	80.1%

DAT_HORA_EVOLUCAO
Categorical

Distinct count	12658
Unique (%)	61.0%
Missing (%)	0.0%
Missing (n)	0

03/10/2018 13:00:00	51
20/06/2018 13:00:00	51
02/07/2018 08:00:00	41
Other values (12655)	20603

Toggle details

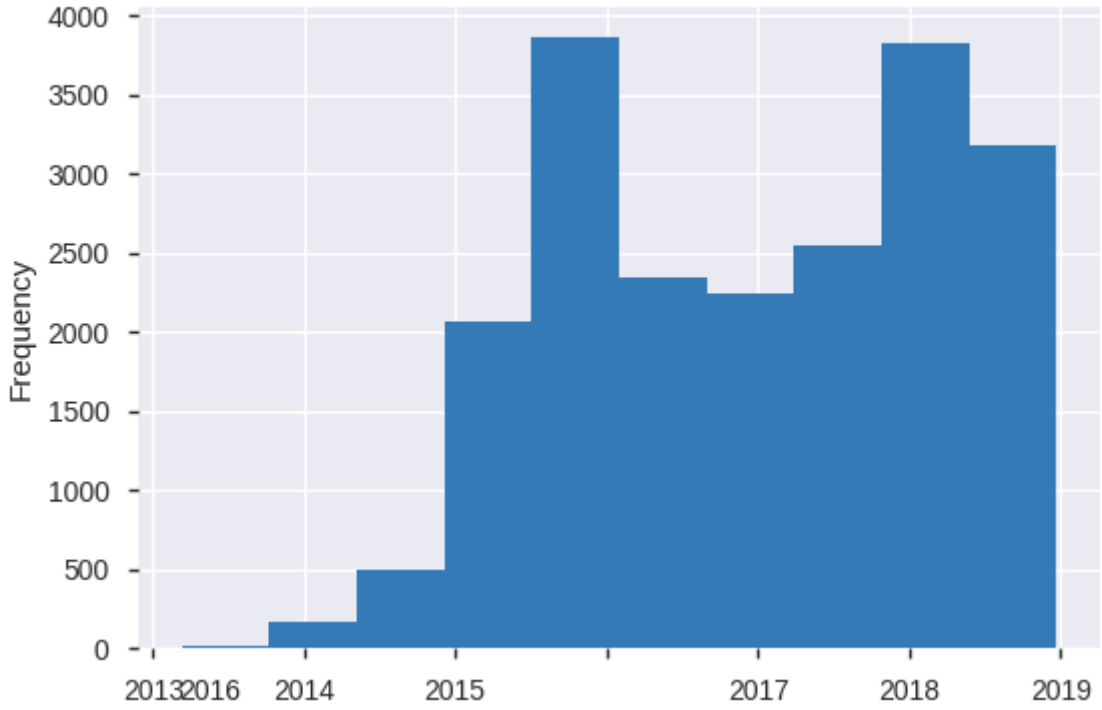
Value	Count	Frequency (%)
03/10/2018 13:00:00	51	0.2%
20/06/2018 13:00:00	51	0.2%
02/07/2018 08:00:00	41	0.2%
02/04/2018 09:00:00	34	0.2%
08/01/2018 09:02:00	28	0.1%
16/04/2018 08:03:00	27	0.1%
17/09/2018 11:08:00	27	0.1%
10/12/2018 09:00:00	19	0.1%

20/08/2018 09:00:00	19	0.1%
30/07/2018 09:00:00	17	0.1%
Other values (12648)	20432	98.5%

DAT_HORA_ANAMNESE
Date

Distinct count	6980
Unique (%)	33.6%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Minimum	2013-03-12 07:00:00
Maximum	2018-12-21 10:00:00

Toggle details



DAT_HORA_ALTA
Categorical

Distinct count	520
Unique (%)	2.5%
Missing (%)	96.8%
Missing (n)	20092

Missing (n)	20092
28/05/2018 11:26:00	11
16/05/2018 15:24:00	6
29/03/2018 13:00:00	5
Other values (516)	632
(Missing)	20092

Toggle details

Value	Count	Frequency (%)
28/05/2018 11:26:00	11	0.1%
16/05/2018 15:24:00	6	0.0%
29/03/2018 13:00:00	5	0.0%
21/05/2018 09:24:00	5	0.0%
22/05/2018 10:45:00	5	0.0%
19/06/2018 10:24:00	4	0.0%
10/05/2018 13:09:00	4	0.0%
30/05/2018 12:11:00	4	0.0%
11/05/2018 08:40:00	3	0.0%
21/03/2018 11:06:00	3	0.0%
Other values (509)	604	2.9%
(Missing)	20092	96.8%

QTD_EVOLUCAO
Numeric

Distinct count	75
Unique (%)	0.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	10.489
Minimum	1
Maximum	102
Zeros (%)	0.0%

Toggle details

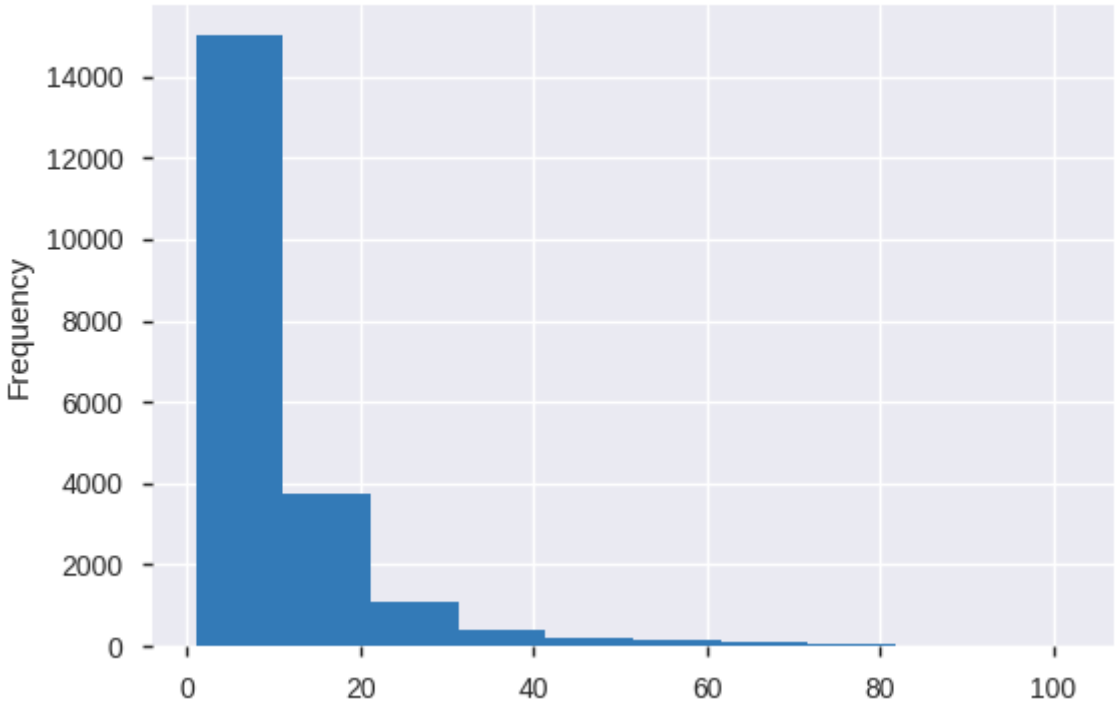
- Statistics
- Histogram
- Common Values
- Extreme Values

Quantile statistics

Minimum	1
5-th percentile	2
Q1	4
Median	8
Q3	12
95-th percentile	30
Maximum	102
Range	101
Interquartile range	8

Descriptive statistics

Standard deviation	10.759
Coef of variation	1.0257
Kurtosis	14.647
Mean	10.489
MAD	6.7367
Skewness	3.2468
Sum	217604
Variance	115.75
Memory size	162.2 KiB



Value	Count	Frequency (%)
4	1692	8.2%
7	1622	7.8%
5	1612	7.8%
3	1587	7.6%
6	1557	7.5%
8	1456	7.0%
9	1334	6.4%
2	1298	6.3%
10	1026	4.9%
11	942	4.5%

Other values (65)	6620	31.9%

Minimum 5 values

Value	Count	Frequency (%)
1	884	4.3%
2	1298	6.3%
3	1587	7.6%
4	1692	8.2%
5	1612	7.8%

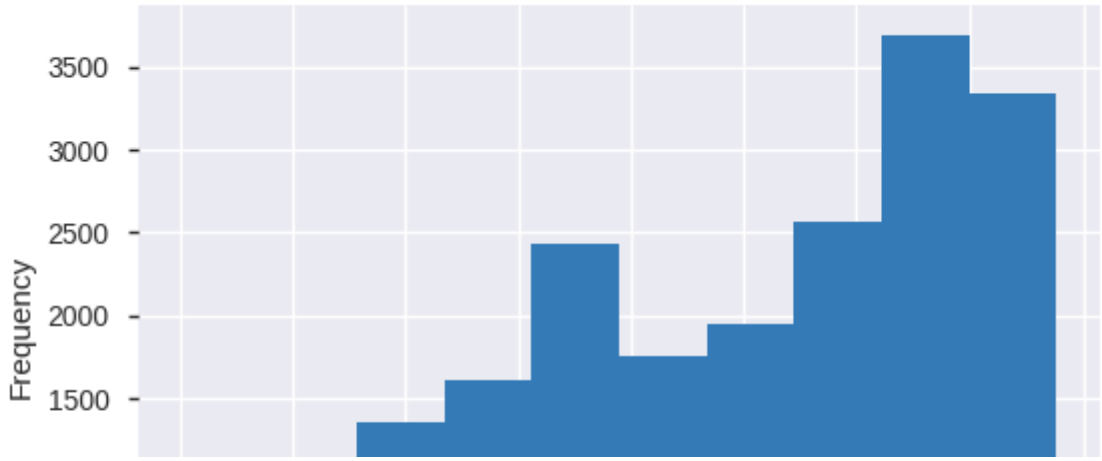
Maximum 5 values

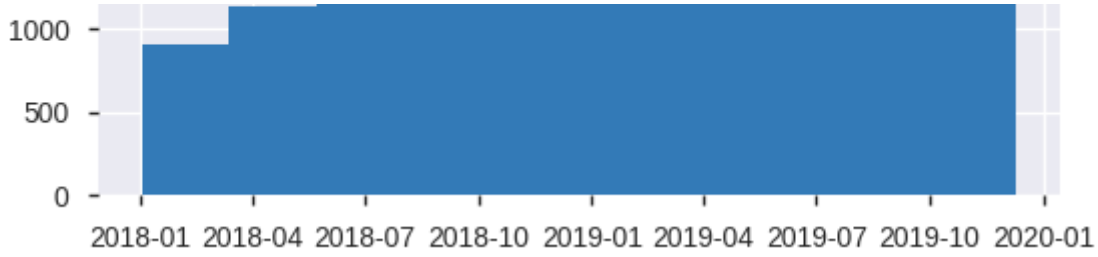
Value	Count	Frequency (%)
78	29	0.1%
83	11	0.1%
89	7	0.0%
95	9	0.0%
102	11	0.1%

DAT_ULTIMA_EVOLUCAO
Date

Distinct count	8975
Unique (%)	43.3%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Minimum	2018-01-02 08:00:00
Maximum	2019-12-09 18:00:00

Toggle details





DIASEMANA
Numeric

Distinct count	7
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	2.1576
Minimum	0
Maximum	6
Zeros (%)	23.8%

Toggle details

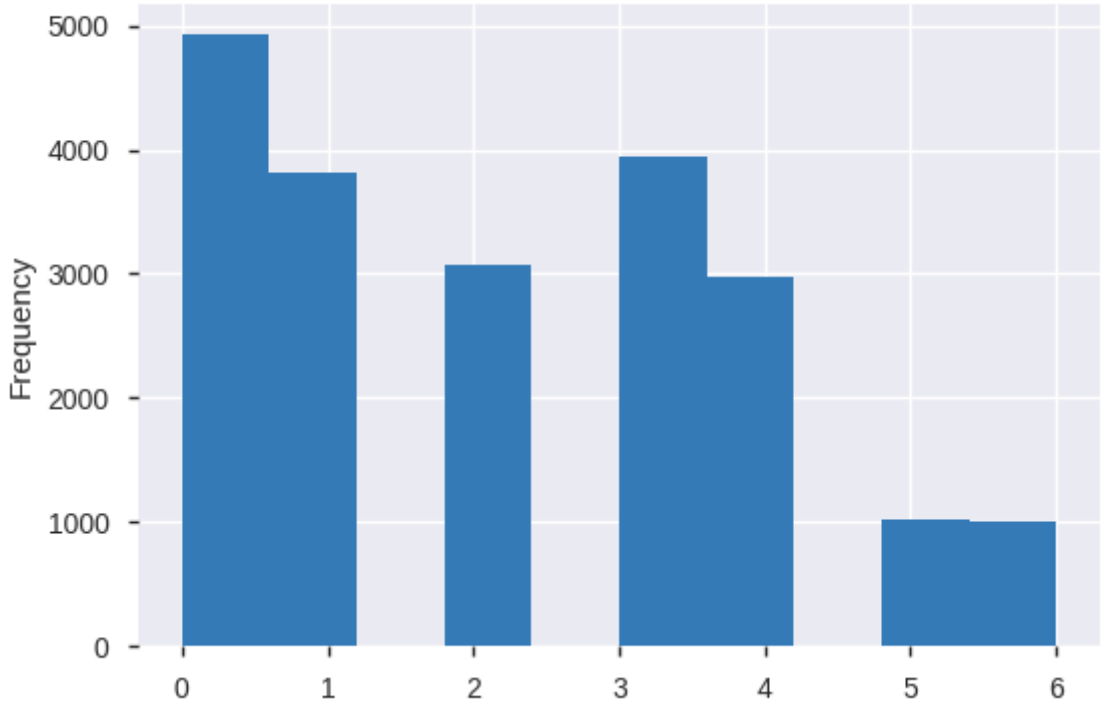
- Statistics
- Histogram
- Common Values
- Extreme Values

Quantile statistics

Minimum	0
5-th percentile	0
Q1	1
Median	2
Q3	3
95-th percentile	5
Maximum	6
Range	6
Interquartile range	2

Descriptive statistics

Standard deviation	1.7563
Coef of variation	0.81398
Kurtosis	-0.80106
Mean	2.1576
MAD	1.4972
Skewness	0.40532
Sum	44762
Variance	3.0844
Memory size	162.2 KiB



Value	Count	Frequency (%)
0	4928	23.8%
3	3942	19.0%
1	3813	18.4%
2	3072	14.8%
4	2974	14.3%
5	1019	4.9%
6	998	4.8%

Minimum 5 values

Value	Count	Frequency (%)
0	4928	23.8%
1	3813	18.4%
2	3072	14.8%
3	3942	19.0%
4	2974	14.3%

Maximum 5 values

Value	Count	Frequency (%)
2	3072	14.8%
3	3942	19.0%
4	2974	14.3%
5	1019	4.9%
6	998	4.8%

PROTOCOLO

Numeric

Distinct count	402
Unique (%)	1.9%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.26415
Minimum	0
Maximum	1.2
Zeros (%)	0.4%

Toggle details

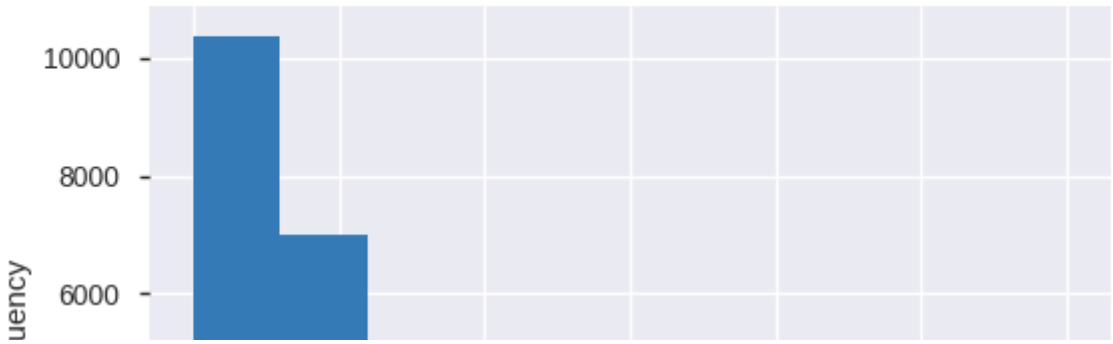
- Statistics
- Histogram
- Common Values
- Extreme Values

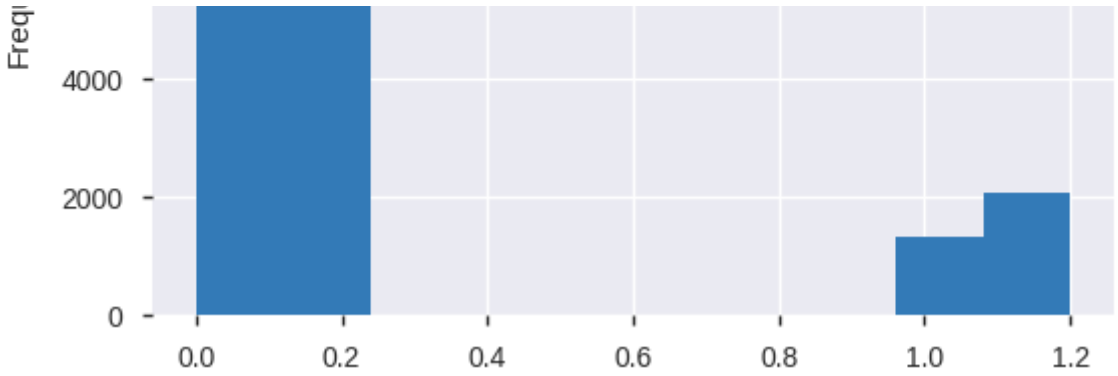
Quantile statistics

Minimum	0
5-th percentile	0.012
Q1	0.06
Median	0.12
Q3	0.18
95-th percentile	1.139
Maximum	1.2
Range	1.2
Interquartile range	0.12

Descriptive statistics

Standard deviation	0.37504
Coef of variation	1.4198
Kurtosis	1.2303
Mean	0.26415
MAD	0.27445
Skewness	1.7499
Sum	5480
Variance	0.14065
Memory size	162.2 KiB





Value	Count	Frequency (%)
0.106	113	0.5%
0.178	108	0.5%
0.046	107	0.5%
0.042	106	0.5%
0.176	104	0.5%
0.129	103	0.5%
0.023	102	0.5%
0.137	101	0.5%
0.093	101	0.5%
0.012	100	0.5%
Other values (392)	19701	95.0%

Minimum 5 values

Value	Count	Frequency (%)
0.0	79	0.4%
0.001	77	0.4%
0.002	93	0.4%
0.003	77	0.4%
0.004	87	0.4%

Maximum 5 values

Value	Count	Frequency (%)
1.196	10	0.0%
1.197	20	0.1%
1.198	14	0.1%
1.199	16	0.1%
1.2	18	0.1%

DURACAO
Categorical

Distinct count	10661
Unique (%)	51.4%
Missing (%)	0.0%
Missing (n)	0

637 days 02:02:00 108

1281 days 06:00:00	98
1393 days 03:51:00	39
Other values (10658)	20501

Toggle details

Value	Count	Frequency (%)
637 days 02:02:00	108	0.5%
1281 days 06:00:00	98	0.5%
1393 days 03:51:00	39	0.2%
1347 days 02:03:00	29	0.1%
1632 days 00:10:00	27	0.1%
970 days 21:57:00	20	0.1%
1191 days 00:56:00	17	0.1%
433 days 03:00:00	16	0.1%
304 days 06:02:00	16	0.1%
694 days 07:00:00	16	0.1%
Other values (10651)	20360	98.1%

NDURACAO

Numeric

Distinct count	10661
Unique (%)	51.4%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	811.13
Minimum	-333
Maximum	2349.2
Zeros (%)	0.0%

Toggle details

- Statistics
- Histogram
- Common Values
- Extreme Values

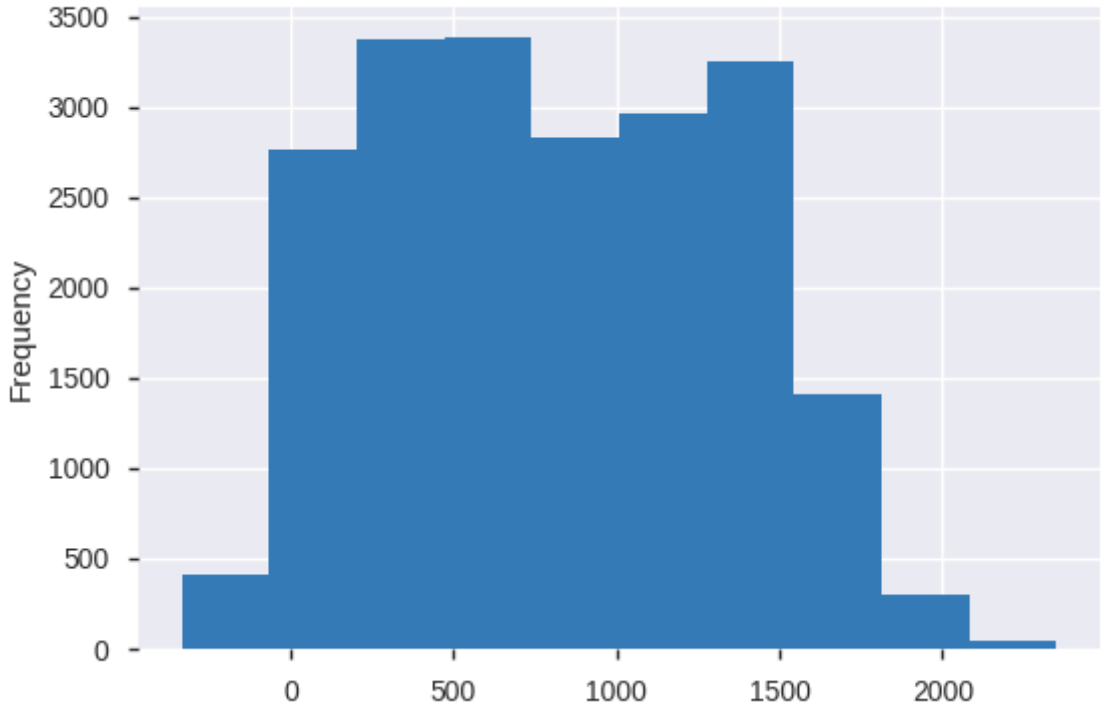
Quantile statistics

Minimum	-333
5-th percentile	35.074
Q1	369.96
Median	783.17

Median	783.17
Q3	1257.1
95-th percentile	1624
Maximum	2349.2
Range	2682.2
Interquartile range	887.19

Descriptive statistics

Standard deviation	524.43
Coef of variation	0.64655
Kurtosis	-1.0262
Mean	811.13
MAD	452.04
Skewness	0.1172
Sum	16828000
Variance	275030
Memory size	162.2 KiB



Value	Count	Frequency (%)
637.0847222222222	108	0.5%
1281.25	98	0.5%
1393.1604166666666	39	0.2%
1347.0854166666666	29	0.1%
1632.0069444444443	27	0.1%
970.9145833333333	20	0.1%
1191.0388888888888	17	0.1%
694.2916666666666	16	0.1%
433.125	16	0.1%
607.9166666666666	16	0.1%
Other values (10651)	20360	98.1%

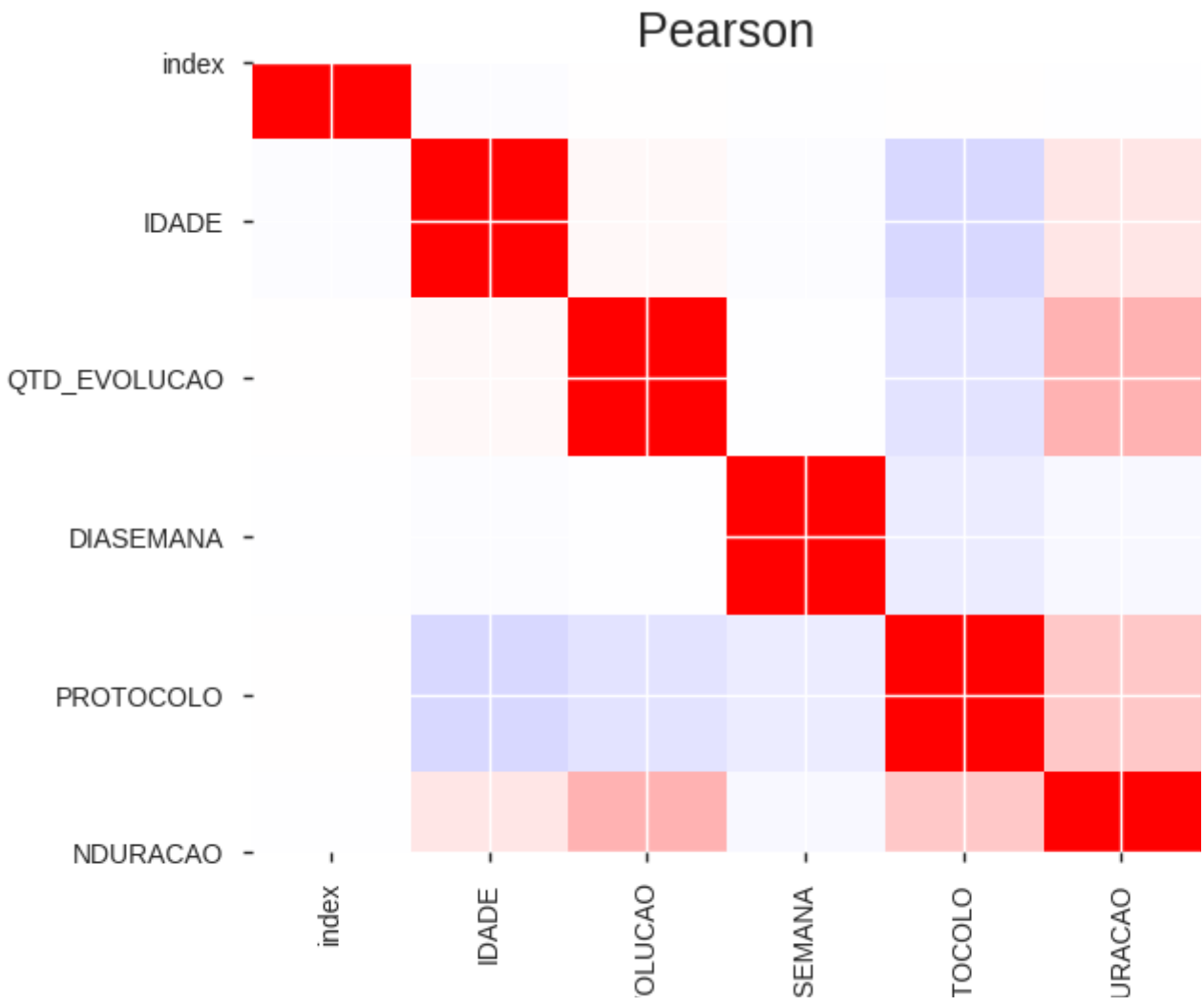
Minimum 5 values

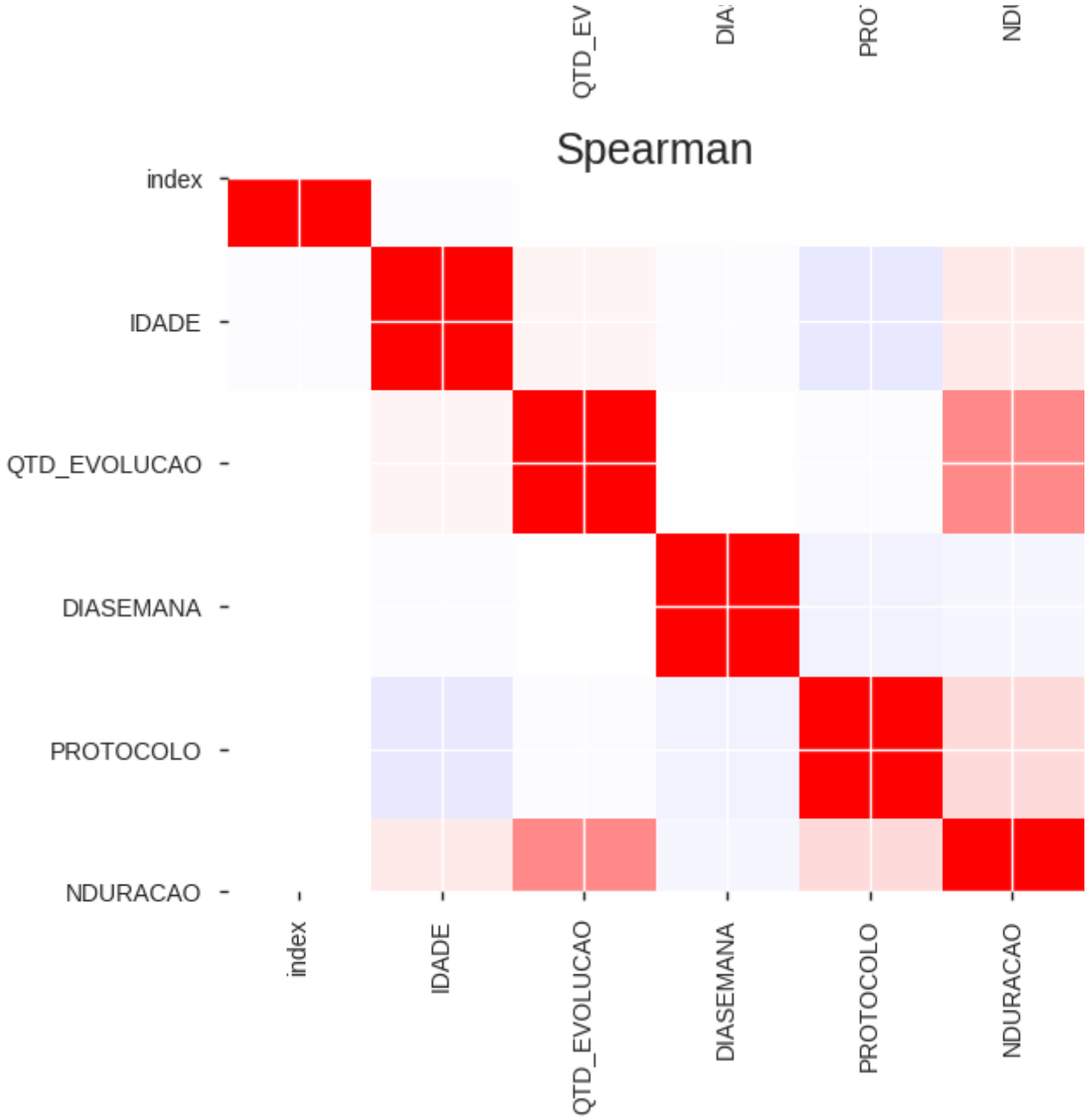
Value	Count	Frequency (%)
-333.0	1	0.0%
-301.80347222222224	1	0.0%
-286.83194444444445	1	0.0%
-286.8229166666667	1	0.0%
-279.87361111111113	2	0.0%

Maximum 5 values

Value	Count	Frequency (%)
2146.0	1	0.0%
2164.085416666667	2	0.0%
2175.0819444444446	2	0.0%
2236.08125	2	0.0%
2349.1652777777776	2	0.0%

Correlations





Sample

	DAT_HORA_ATENDIMENTO	NOM_ENCAMINHAMENTO	NOM_MODALIDADE_ATENDIMENTO	NOI
3	2018-10-22 07:12:00	RETORNO	AMBULATORIO	
7	2018-05-22 12:00:00	RETORNO	AMBULATORIO	
8	2018-07-24 07:00:00	RETORNO	AMBULATORIO	

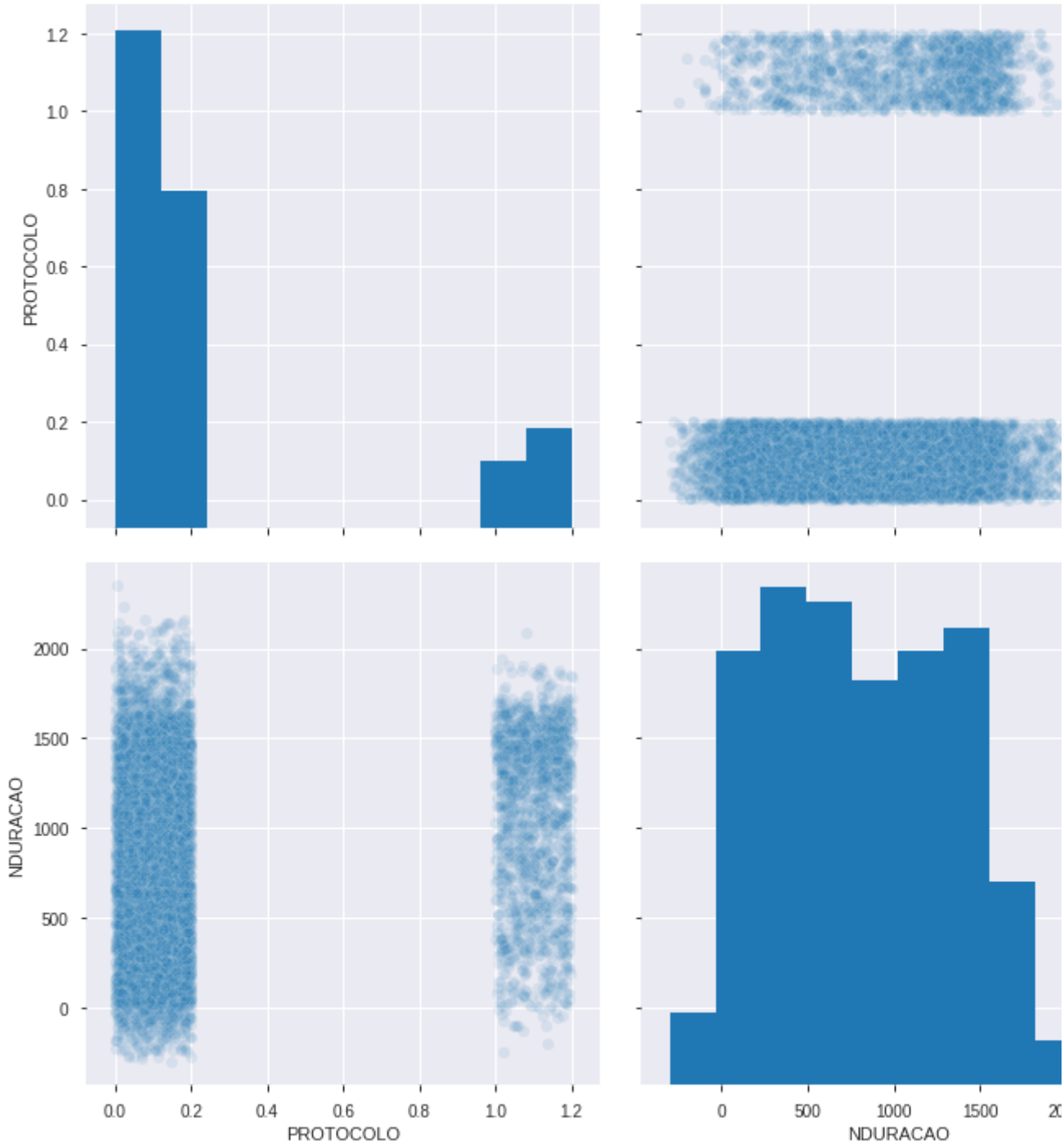
10	2018-04-12 07:11:00	RETORNO	AMBULATORIO	DC
14	2018-11-19 07:00:00	RETORNO	AMBULATORIO	

Análises

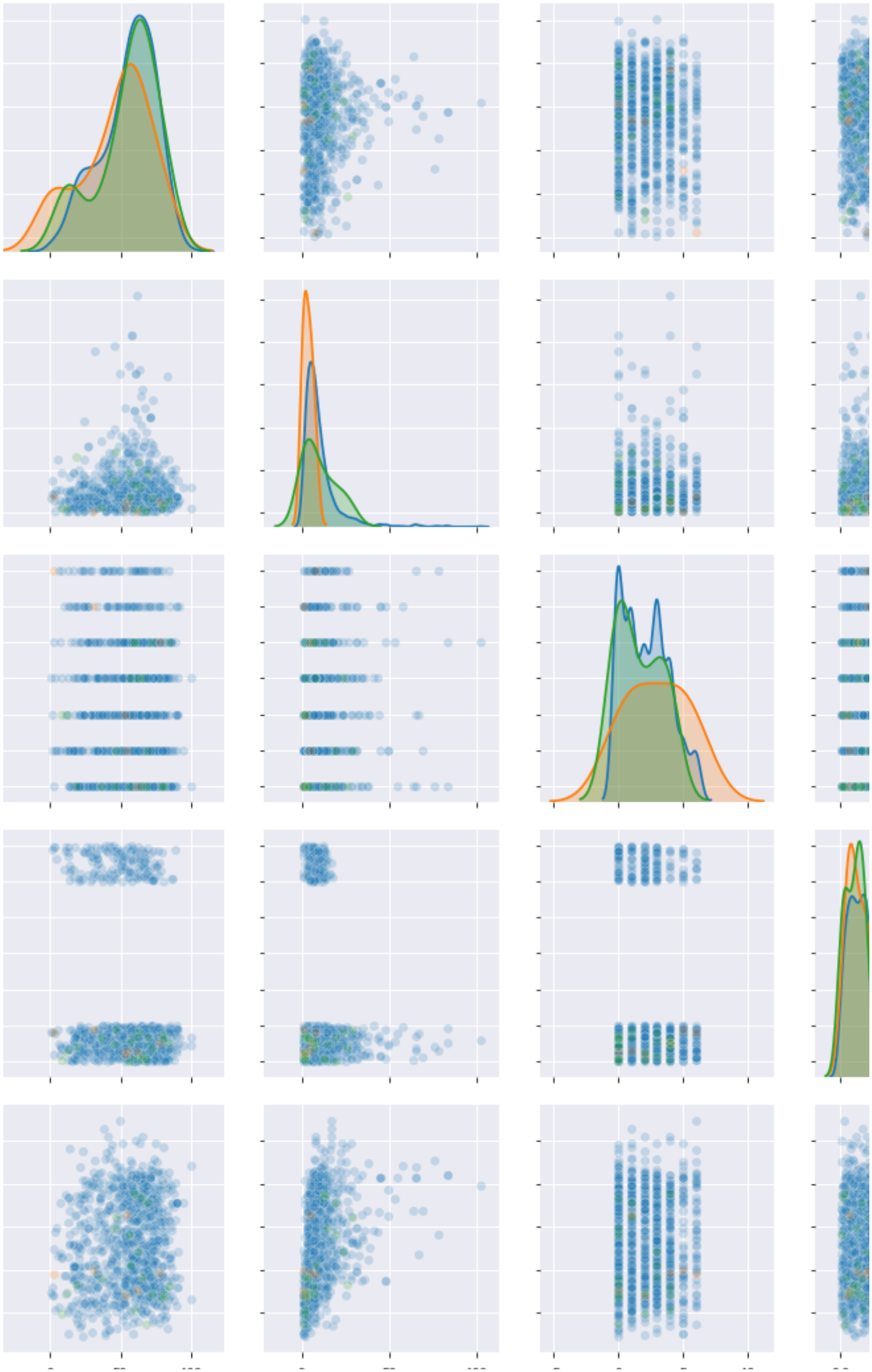
Plot do dataset puro

Protocolo x Duração

```
dfProtocoloDuracao = dfLimpo[['PROTOCOLO','NDURACAO']].sample(10000)
%matplotlib inline
sb.pairplot(dfProtocoloDuracao,height=5,kind='scatter', plot_kws={'alpha':0.1})
pl.show()
```




```
%matplotlib inline
sb.pairplot(dfLimpo.sample(1000),hue='NOM_MODALIDADE_ATENDIMENTO',height=3,kind='scatter',
pl.show()
```




```
dfCluster = dfLimpo[['IDADE','PROTOCOLO','NDURACAO']]
X = np.array(dfCluster)
```

Clusterização

```
from sklearn.cluster import KMeans
```

```
kmeans = KMeans( n_clusters=2 ,random_state=0)
```

```
dfCluster
```

	IDADE	PROTOCOLO	NDURACAO	cluster
3	51.941257	0.176	1233.008333	3
7	54.089202	0.157	465.290972	2
8	60.264544	0.105	309.329861	4
10	33.530298	0.093	-214.965278	0
14	57.154955	0.196	449.392361	2
...	...	...	...	...
49172	58.308380	1.145	1149.045833	1
49178	74.020709	1.080	1729.039583	5
49179	60.056325	0.141	1335.247222	3
49182	77.390572	0.116	522.086806	2
49183	67.998791	1.040	708.041667	6

20746 rows × 4 columns

```
kmeans.fit(X)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=0, tol=0.0001, verbose=0)
```

```
kmeans.labels_
```

```
array([0, 1, 1, ..., 0, 1, 1], dtype=int32)
```

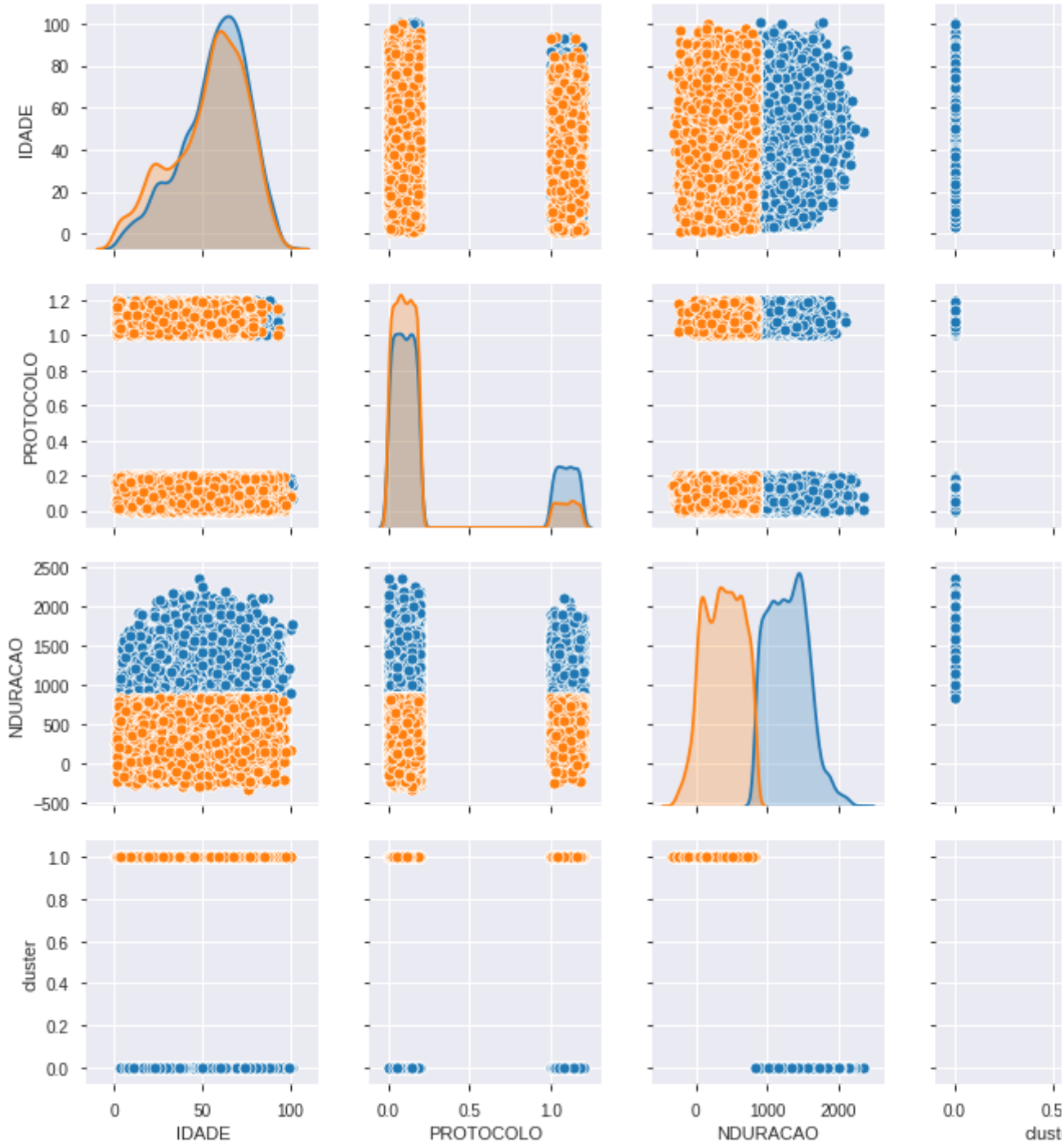
```
dfCluster['cluster'] = kmeans.labels
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarnir
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/use
  """Entry point for launching an IPython kernel.
```

Plotando o resultado da clusterização

```
sb.pairplot(dfCluster,hue='cluster')
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/nonparametric/kde.py:487: RuntimeW
  binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
/usr/local/lib/python3.6/dist-packages/statsmodels/nonparametric/kdetools.py:34: Runt
  FAC1 = 2*(np.pi*bw/RANGE)**2
<seaborn.axisgrid.PairGrid at 0x7fcfc2adf208>
```



Regressão

Double-click (or enter) to edit

```
dfLimpo
```

	DAT_HORA_ATENDIMENTO	NOM_ENCAMINHAMENTO	NOM_MODALIDADE_ATENDIMENTO	NOM_MUN
3	2018-10-22 07:12:00	RETORNO	AMBULATORIO	
7	2018-05-22 12:00:00	RETORNO	AMBULATORIO	L
8	2018-07-24 07:00:00	RETORNO	AMBULATORIO	PED
10	2018-04-12 07:11:00	RETORNO	AMBULATORIO	DOMIN MAR
14	2018-11-19 07:00:00	RETORNO	AMBULATORIO	FOR
...	...	...	...	
49172	2018-07-30 12:06:00	RETORNO	AMBULATORIO	
49178	2018-07-05 07:00:00	RETORNO	AMBULATORIO	O
49179	2018-04-10 12:00:00	RETORNO	AMBULATORIO	CR
49182	2018-05-04 12:08:00	RETORNO	AMBULATORIO	PARA PA
49183	2018-02-19 07:10:00	RETORNO	AMBULATORIO	O

20746 rows × 18 columns

```
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

dfRegressao = dfLimpo[['NDURACAO','PROTOCOLO','NOM_MODALIDADE_ATENDIMENTO','QTD_EVOLUCAO']

dfRegressao = pd.concat([dfRegressao, pd.get_dummies(dfRegressao['NOM_MODALIDADE_ATENDIMEN

dfRegressao
```

	NDURACAO	PROTOCOLO	NOM_MODALIDADE_ATENDIMENTO	QTD_EVOLUCAO	AMBULATORIO
3	1233.008333	0.176	AMBULATORIO	9	1
7	465.290972	0.157	AMBULATORIO	4	1
8	309.329861	0.105	AMBULATORIO	3	1
10	-214.965278	0.093	AMBULATORIO	1	1
14	449.392361	0.196	AMBULATORIO	5	1
...	...	...	...	...	...
49172	1149.045833	1.145	AMBULATORIO	14	1
49178	1729.039583	1.080	AMBULATORIO	8	1
49179	1335.247222	0.141	AMBULATORIO	16	1
49182	522.086806	0.116	AMBULATORIO	2	1
49183	708.041667	1.040	AMBULATORIO	6	1

20746 rows × 8 columns

```
dfRegressao = dfRegressao.drop('NOM_MODALIDADE_ATENDIMENTO', axis=1)

# passando os valores de x e y como Dataframes
X = dfRegressao[['PROTOCOLO','AMBULATORIO','INTERNAÇÃO','SADT EXTERNO','SADT UBS MARILIA',
Y = dfRegressao[['NDURACAO']]
# criando e treinando o modelo
model = LinearRegression()
model.fit(X, Y)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

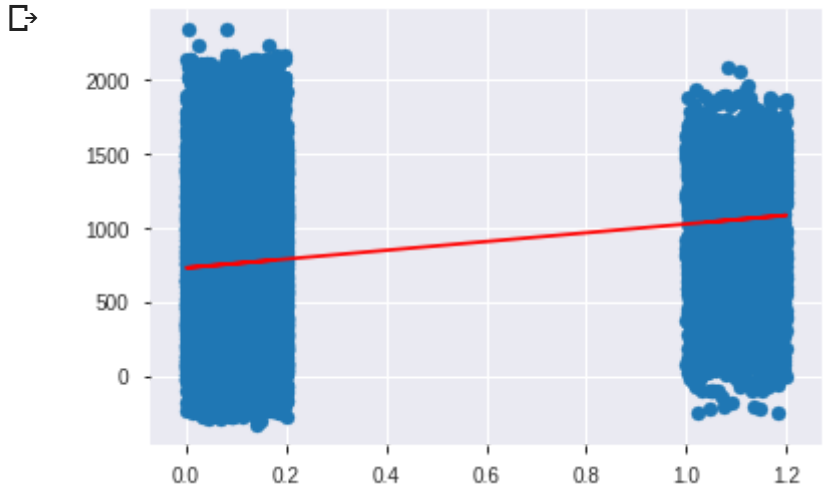
Teste predicao regressao

```
teste = [[0,1,0,0,0,1]]

model.predict(teste)
```

```
array([[570.87413413]])
```

Plot regressao

```
%matplotlib inline
# passando os valores de x e y como Dataframes
dfRegressaoPlot = dfRegressao
X = dfRegressaoPlot[['PROTOCOLO']]
Y = dfRegressaoPlot[['NDURACAO']]
# criando e treinando o modelo
model = LinearRegression()
model.fit(X, Y)
Y_pred = model.predict(X)
pl.scatter(X, Y)


pl.plot(X, Y_pred, color='red')
pl.show()
```



Correção dos OUTLIERS

Double-click (or enter) to edit

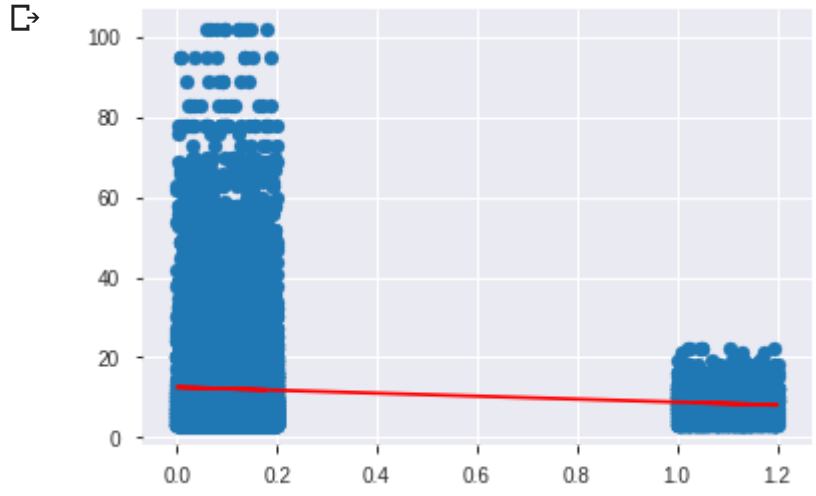
```
%matplotlib inline
# passando os valores de x e y como Dataframes

dfRegressaoCorrigido = dfLimpo[['NDURACAO','PROTOCOLO','NOM_MODALIDADE_ATENDIMENTO','QTD_E

dfRegressaoCorrigido = pd.concat([dfRegressaoCorrigido, pd.get_dummies(dfRegressaoCorrigic

dfRegressaoPlot = dfRegressaoCorrigido[dfRegressaoCorrigido.QTD_EVOLUCAO>2]

dfRegressaoPlot
```

	NDURACAO	PROTOCOLO	NOM_MODALIDADE_ATENDIMENTO	QTD_EVOLUCAO	AMBULATORIO
3	1233.008333	0.176	AMBULATORIO	9	1
7	465.290972	0.157	AMBULATORIO	4	1
8	309.329861	0.105	AMBULATORIO	3	1
14	449.392361	0.196	AMBULATORIO	5	1
17	139.040278	0.036	AMBULATORIO	3	1
...	...	...	...	...	...
49169	1224.047917	0.050	AMBULATORIO	10	1
49172	1149.045833	1.145	AMBULATORIO	14	1
49178	1729.039583	1.080	AMBULATORIO	8	1
49179	1335.247222	0.141	AMBULATORIO	16	1
49183	708.041667	1.040	AMBULATORIO	6	1

18564 rows × 8 columns

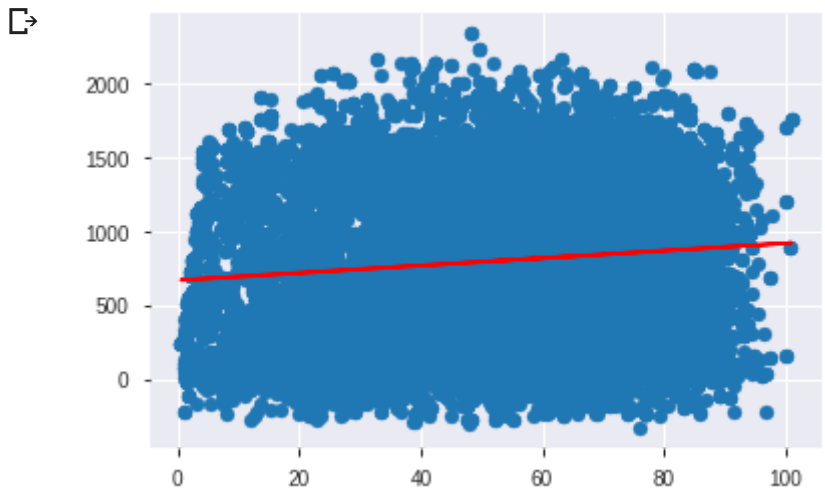
```
X = dfRegressaoPlot[['PROTOCOLO']]
Y = dfRegressaoPlot[['QTD_EVOLUCAO']]
# criando e treinando o modelo
model = LinearRegression()
model.fit(X, Y)
Y_pred = model.predict(X)
pl.scatter(X, Y)
```

```
pl.plot(X, Y_pred, color='red')
pl.show()
```



Regressão com IDADE

```
%matplotlib inline
# passando os valores de x e y como Dataframes

dfRegressaoCorrigido = dfLimpo[['IDADE','NDURACAO']]



X = dfRegressaoCorrigido[['IDADE']]
Y = dfRegressaoCorrigido[['NDURACAO']]
# criando e treinando o modelo
model = LinearRegression()
model.fit(X, Y)
Y_pred = model.predict(X)
pl.scatter(X, Y)


pl.plot(X, Y_pred, color='red')
pl.show()
```




```
dfLimpo
```

TESTE regressão idade

```
#15 anos
teste = [[15]]

model.predict(teste)
```

```
array([[709.41687913]])
```

```
#70 anos
teste = [[65]]

model.predict(teste)
```

```
array([[834.80425704]])
```

''

Conclusão

Foi CONSTATADO que a eficiência da especialidade está intimamente ligada à aplicação correta protocolo.