Problema

Existe uma hipótese de que o referenciamento de pacientes ao HC não é totalmente necessário. I
tratado na UBS, pois o HC é um hospital de alta complexidade.

Hipótese desta análise(opcional)

Esta analise tem uma hipotese de que a efetividade do tratamento de uma equipe esta correlacio

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: 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: PyYAML>=3.10 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: numpy>=1.7.1 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: tornado>=4.3 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: pillow>=4.0 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6/dist-pack
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
```

```
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
UROLÓGIA                                25
GENÉTICA                                25
IMUNOPATOLOGIA CLÍNICA E ALÉRGICA       20
ONCOCLÍNICA                             18
PRÉ-OPERATÓRIO                          13
PSICOLOGIA HOSPITALAR                    6
CLÍNICA MÉDICA ESPECIALIZADA             5
ENFERMAGEM                               3
CENTRO CIRÚRGICO                         2
HEMOTERAPIA                              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 SASC
```

```
dfLimpo["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
Name: NOM_EQUIPE, 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['DURACAO'] = dfLimpo['DAT_ULTIMA_EVOLUCAO'].sub(dfLimpo['DAT_HORA_ANAMNESE'], axis
dfLimpo['NDURACAO'] = dfLimpo['DURACAO'] / np.timedelta64(1, 'D')
dfFiltro = dfLimpo[dfLimpo.NOM_EQUIPE=='REUMATOLOGIA']

dfFiltro
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:11: SettingWithCopyWarni
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
  # This is added back by InteractiveShellApp.init_path()
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:12: SettingWithCopyWarni
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
  if sys.path[0] == '':
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:13: SettingWithCopyWarni
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
  del sys.path[0]
```

	DAT_HORA_ATENDIMENTO	NOM_ENCAMINHAMENTO	NOM_MODALIDADE_ATENDIMENTO	NOM_MUNICIPI

```
dfFiltro
```

	DAT_HORA_ATENDIMENTO	NOM_ENCAMINHAMENTO	NOM_MODALIDADE_ATENDIMENTO	NOM_MUNICIPI

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	32012
Total Missing (%)	9.7%
Total size in memory	4.6 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: 1210 distinct values Warning
- COD_CID has a high cardinality: 1347 distinct values Warning
- DAT_HORA_PREVISTA has 27869 / 87.1% missing values Missing
- DAT_HORA_PREVISTA has a high cardinality: 1558 distinct values Warning
- DAT_HORA_EVOLUCAO has a high cardinality: 16067 distinct values Warning
- DAT_HORA_ALTA has 31266 / 97.7% missing values Missing
- DAT_HORA_ALTA has a high cardinality: 596 distinct values Warning
- DIASEMANA has 6848 / 21.4% zeros Zeros
- DURACAO has a high cardinality: 13369 distinct values Warning

Variables

index

Numeric

Distinct count	32012
Unique (%)	100.0%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	24525
Minimum	1
Maximum	49186

Zeros (%)	0.0%

Toggle details

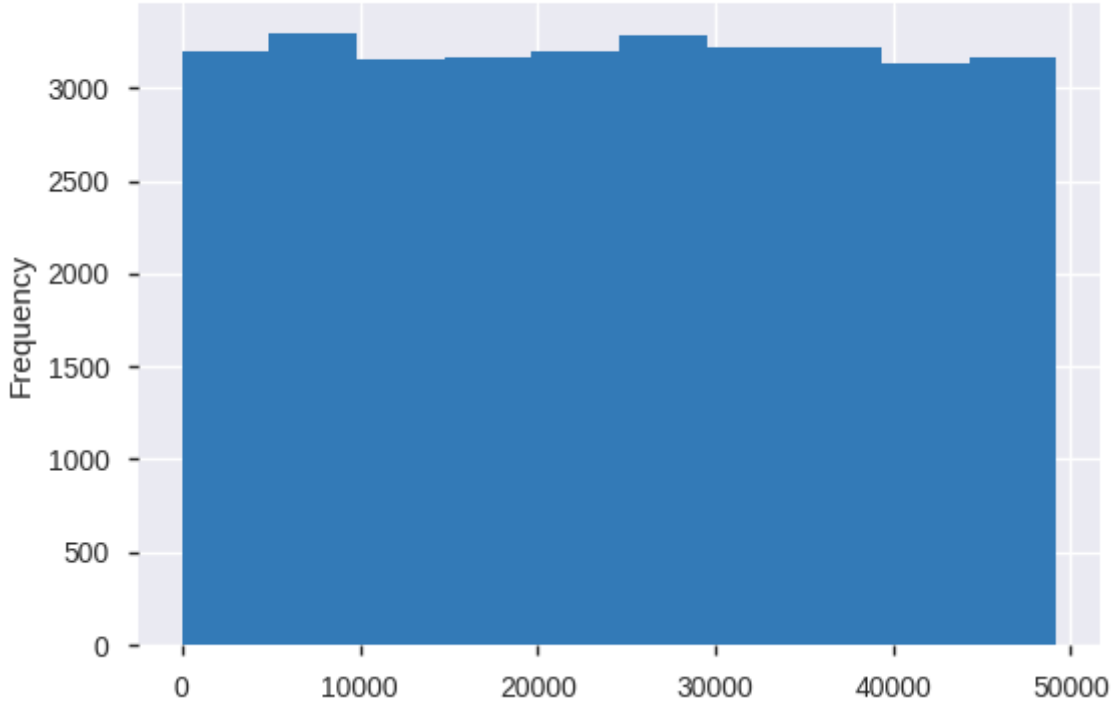
- Statistics
- Histogram
- Common Values
- Extreme Values

Quantile statistics

Minimum	1
5-th percentile	2377.1
Q1	12236
Median	24614
Q3	36713
95-th percentile	46666
Maximum	49186
Range	49185
Interquartile range	24476

Descriptive statistics

Standard deviation	14178
Coef of variation	0.57812
Kurtosis	-1.1956
Mean	24525
MAD	12272
Skewness	-0.00025111
Sum	785094441
Variance	201030000
Memory size	250.2 KiB



Value	Count	Frequency (%)
2047	1	0.0%
11503	1	0.0%
48349	1	0.0%
42206	1	0.0%
23777	1	0.0%
17634	1	0.0%
19683	1	0.0%
31973	1	0.0%
27879	1	0.0%
5352	1	0.0%
Other values (32002)	32002	100.0%

Minimum 5 values

Value	Count	Frequency (%)
1	1	0.0%
3	1	0.0%
4	1	0.0%
5	1	0.0%
6	1	0.0%

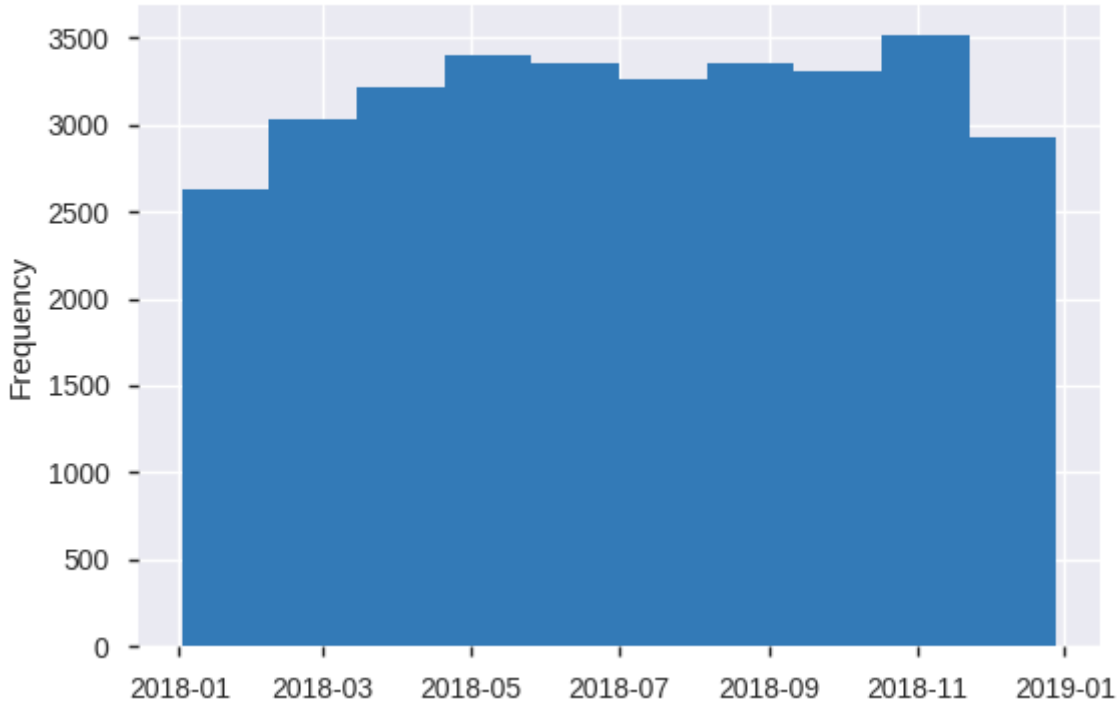
Maximum 5 values

Value	Count	Frequency (%)
49178	1	0.0%
49179	1	0.0%
49182	1	0.0%
49183	1	0.0%
49186	1	0.0%

DAT_HORA_ATENDIMENTO
Date

Distinct count	10205
Unique (%)	31.9%
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	31
Unique (%)	0.1%
Missing (%)	0.0%
Missing (n)	0

RETORNO	30752
ALTA	770
ALTA E ENCAMINHAMENTO	114
Other values (28)	376

Toggle details

Value	Count	Frequency (%)
RETORNO	30752	96.1%
ALTA	770	2.4%
ALTA E ENCAMINHAMENTO	114	0.4%
RETORNO E ENCAMINHAMENTO	89	0.3%
AGUARDANDO CIRURGIA	80	0.2%
PEDIDO DE INTERNAÇÃO HC-I	72	0.2%
CONTRA-REFERENCIA	48	0.1%
FALTA A CONSULTA AGENDADA	25	0.1%
ENCAM.UBS/PSF DE ORIGEM	9	0.0%
ENCAM.AMBULATORIO HC	7	0.0%

Other values (21)	46	0.1%

NOM_MODALIDADE_ATENDIMENTO
Categorical

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

AMBULATORIO	31361
INTERNAÇÃO	397
SADT EXTERNO	187

Toggle details

Value	Count	Frequency (%)
AMBULATORIO	31361	98.0%
INTERNAÇÃO	397	1.2%
SADT EXTERNO	187	0.6%
SADT UBS MARILIA	67	0.2%

NOM_MUNICIPIO
Categorical

Distinct count	1210
Unique (%)	3.8%
Missing (%)	0.0%
Missing (n)	0

MARILIA	10556
GARÇA	1250
VERA CRUZ	1042
Other values (1207)	19164

Toggle details

Value	Count	Frequency (%)
MARILIA	10556	33.0%
GARÇA	1250	3.9%
VERA CRUZ	1042	3.3%
TUPÃ	783	2.4%
SÃO PAULO	774	2.4%

POMPÉIA	704	2.2%
GÁLIA	641	2.0%
LUPÉRCIO	568	1.8%
OCAUÇU	488	1.5%
ORIENTE	461	1.4%
Other values (1200)	14745	46.1%

NOM_EQUIPE
Categorical

Distinct count	8
Unique (%)	0.0%
Missing (%)	0.0%
Missing (n)	0

AMBULATÓRIO SAÚDE MENTAL	11266
ORTOPEDIA E TRAUMATOLOGIA	4210
OFTALMOLOGIA	4049
Other values (5)	12487

Toggle details

Value	Count	Frequency (%)
AMBULATÓRIO SAÚDE MENTAL	11266	35.2%
ORTOPEDIA E TRAUMATOLOGIA	4210	13.2%
OFTALMOLOGIA	4049	12.6%
ENDOCRINOLOGIA E METABOLISMO	3404	10.6%
NEUROLOGIA	2410	7.5%
CIRURGIA VASCULAR	2374	7.4%
ONCOLOGIA CLÍNICA	2268	7.1%
DERMATOLOGIA	2031	6.3%

NOM_TIPO_CASO
Categorical

Distinct count	39
Unique (%)	0.1%
Missing (%)	0.0%
Missing (n)	0

RETORNO	19823
AGENDADO PELO PROFISSIONAL	2949

AGENDADO	2941
Other values (36)	6299

Toggle details

Value	Count	Frequency (%)
RETORNO	19823	61.9%
AGENDADO PELO PROFISSIONAL	2949	9.2%
AGENDADO	2941	9.2%
RETORNO FALTOSOS	1258	3.9%
QUIMIOTERAPIA	983	3.1%
ENCAIXE AUTORIZADO	887	2.8%
RETORNO MÉDICO	677	2.1%
REGULAÇÃO INTERNA	527	1.6%
NOVO	467	1.5%
CONSULTA INICIAL NO PROGRAMA	254	0.8%
Other values (29)	1246	3.9%

IDADE
Numeric

Distinct count	10315
Unique (%)	32.2%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	51.648
Minimum	0.60153
Maximum	100.94
Zeros (%)	0.0%

Toggle details

- Statistics
- Histogram
- Common Values
- Extreme Values

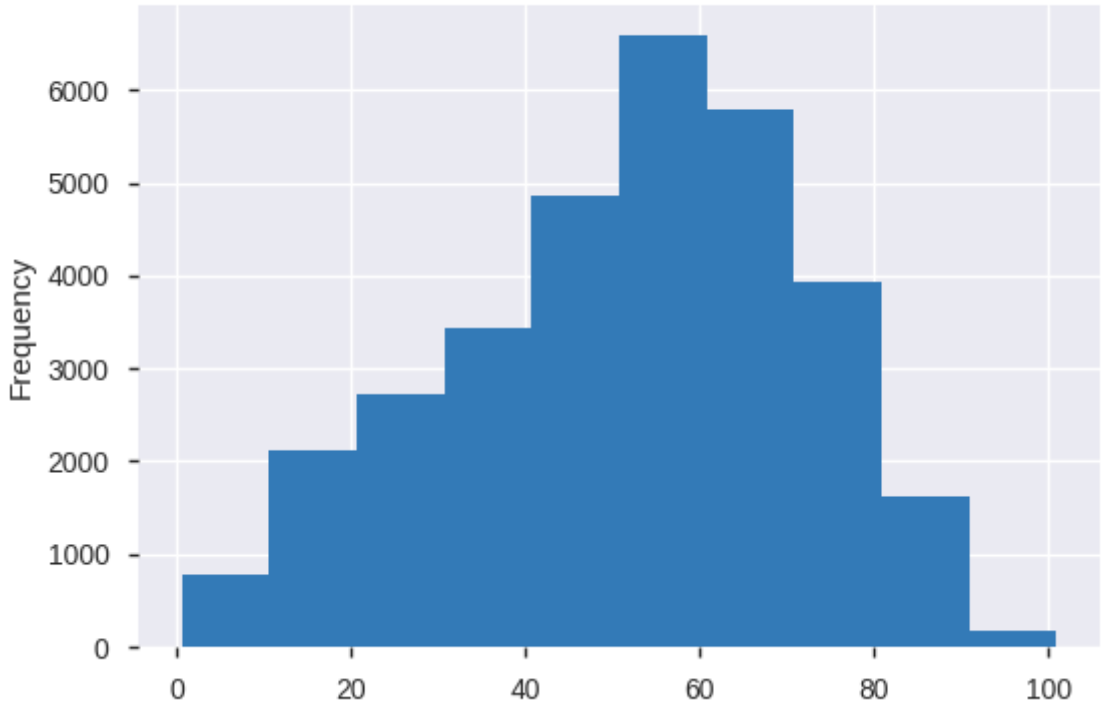
Quantile statistics

Minimum	0.60153
5-th percentile	14.87
Q1	38.212

Median	53.971
Q3	66.51
95-th percentile	81.373
Maximum	100.94
Range	100.34
Interquartile range	28.298

Descriptive statistics

Standard deviation	20.08
Coef of variation	0.38878
Kurtosis	-0.55158
Mean	51.648
MAD	16.397
Skewness	-0.3462
Sum	1653400
Variance	403.2
Memory size	250.2 KiB



Value	Count	Frequency (%)
58.946735984271896	108	0.3%
90.15495516235409	98	0.3%
70.9165989979706	46	0.1%
24.6755031075596	36	0.1%
56.0426263952308	32	0.1%
29.270023655504797	28	0.1%
82.5604346144089	28	0.1%
45.322078450025394	27	0.1%
50.2207085870117	24	0.1%
57.2289277650939	23	0.1%
Other values (10305)	31562	98.6%

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

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	1347
Unique (%)	4.2%
Missing (%)	0.0%
Missing (n)	0

Z988	2045	
F200	1440	
Z010	1290	
Other values (1344)		27237

Toggle details

Value	Count	Frequency (%)
Z988	2045	6.4%
F200	1440	4.5%
Z010	1290	4.0%
F603	1284	4.0%
F604	923	2.9%
L989	786	2.5%
Z000	742	2.3%
E119	739	2.3%
H409	615	1.9%
F411	570	1.8%
Other values (1337)	21578	67.4%

DAT_HORA_PREVISTA
Categorical

Distinct count	1558
Unique (%)	4.9%
Missing (%)	87.1%
Missing (n)	27869

23/04/2019 07:00:00	41
11/03/2019 07:00:00	25
22/01/2019	21
Other values (1554)	4056
(Missing)	27869

Toggle details

Value	Count	Frequency (%)
23/04/2019 07:00:00	41	0.1%
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%
23/01/2019 07:00:00	18	0.1%
18/03/2019	18	0.1%
23/04/2019 12:00:00	18	0.1%
15/04/2019 07:00:00	18	0.1%
Other values (1547)	3926	12.3%
(Missing)	27869	87.1%

DAT_HORA_EVOLUCAO
Categorical

Distinct count	16067
Unique (%)	50.2%
Missing (%)	0.0%
Missing (n)	0

20/06/2018 13:00:00	54
03/10/2018 13:00:00	53
02/07/2018 08:00:00	44
Other values (16064)	31861

Toggle details

Value	Count	Frequency (%)

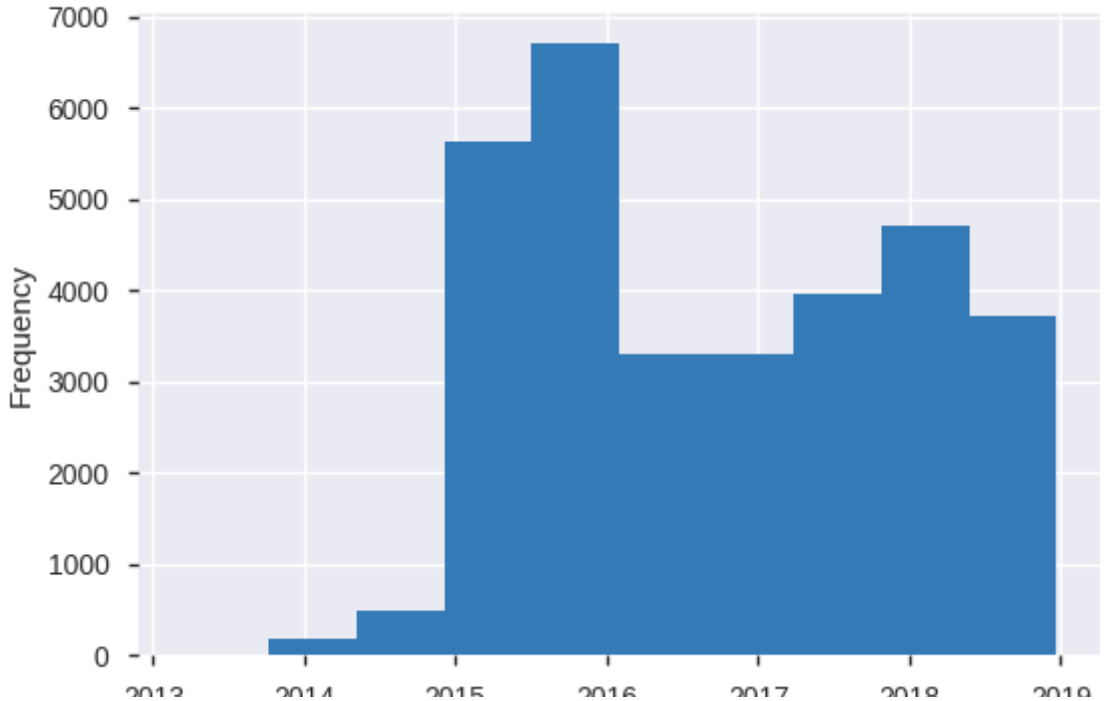
Value	Count	Frequency (%)
20/06/2018 13:00:00	54	0.2%
03/10/2018 13:00:00	53	0.2%
02/07/2018 08:00:00	44	0.1%
02/04/2018 09:00:00	35	0.1%
08/01/2018 09:02:00	28	0.1%
16/04/2018 08:03:00	27	0.1%
17/09/2018 11:08:00	27	0.1%
04/05/2018 10:00:00	21	0.1%
03/05/2018 11:00:00	20	0.1%
10/12/2018 09:00:00	20	0.1%
Other values (16057)	31683	99.0%

DAT_HORA_ANAMNESE

Date

Distinct count	9059
Unique (%)	28.3%
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



2013 2014 2015 2016 2017 2018 2019

DAT_HORA_ALTA
Categorical

Distinct count	596
Unique (%)	1.9%
Missing (%)	97.7%
Missing (n)	31266

28/05/2018 11:26:00	11	
16/05/2018 15:24:00	6	
29/03/2018 13:00:00	5	
Other values (592)	724	
(Missing)		31266

Toggle details

Value	Count	Frequency (%)
28/05/2018 11:26:00	11	0.0%
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%
10/05/2018 13:09:00	4	0.0%
19/06/2018 10:24:00	4	0.0%
30/05/2018 12:11:00	4	0.0%
06/04/2018 08:14:00	3	0.0%
11/05/2018 09:14:00	3	0.0%
Other values (585)	696	2.2%
(Missing)	31266	97.7%

QTD_EVOLUCAO
Numeric

Distinct count	75
Unique (%)	0.2%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	12.765
Minimum	1
Maximum	102
Zeros (%)	0.0%

Toggle details

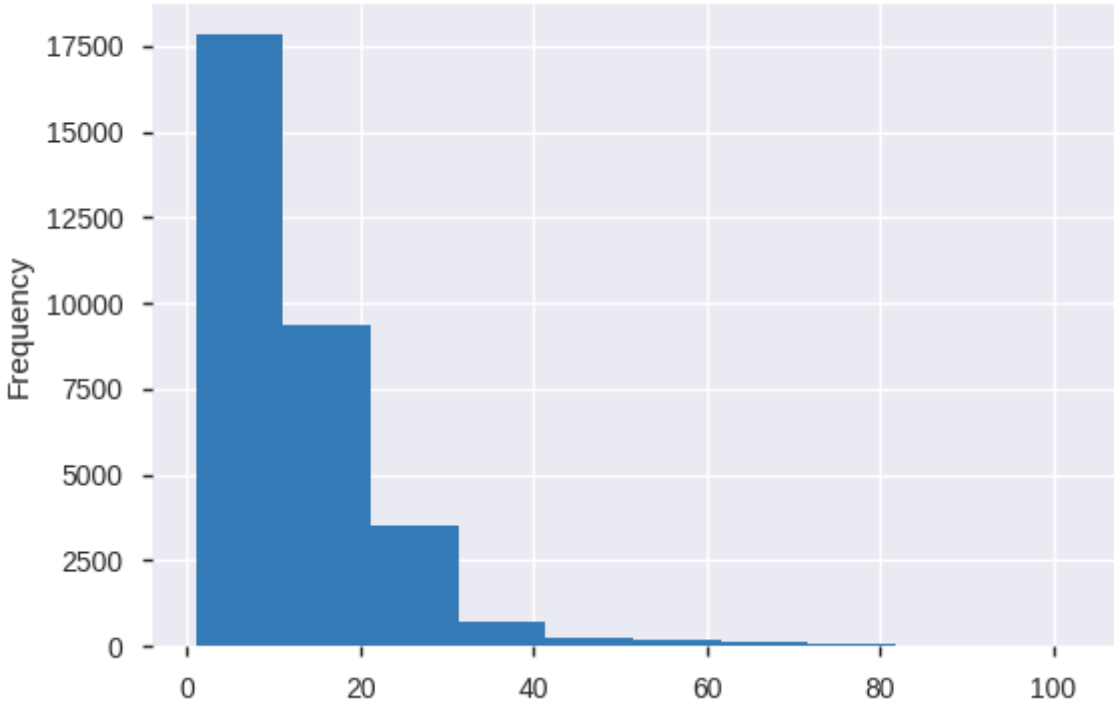
- Statistics
- Histogram
- Common Values
- Extreme Values

Quantile statistics

Minimum	1
5-th percentile	2
Q1	6
Median	10
Q3	18
95-th percentile	30
Maximum	102
Range	101
Interquartile range	12

Descriptive statistics

Standard deviation	10.215
Coef of variation	0.8002
Kurtosis	9.9347
Mean	12.765
MAD	7.3757
Skewness	2.3187
Sum	408636
Variance	104.34
Memory size	250.2 KiB



Value	Count	Frequency (%)
7	1853	5.8%

4	1822	5.7%
6	1817	5.7%
8	1772	5.5%
5	1759	5.5%
3	1711	5.3%
9	1704	5.3%
11	1543	4.8%
10	1516	4.7%
2	1389	4.3%
Other values (65)	15126	47.3%

Minimum 5 values

Value	Count	Frequency (%)
1	939	2.9%
2	1389	4.3%
3	1711	5.3%
4	1822	5.7%
5	1759	5.5%

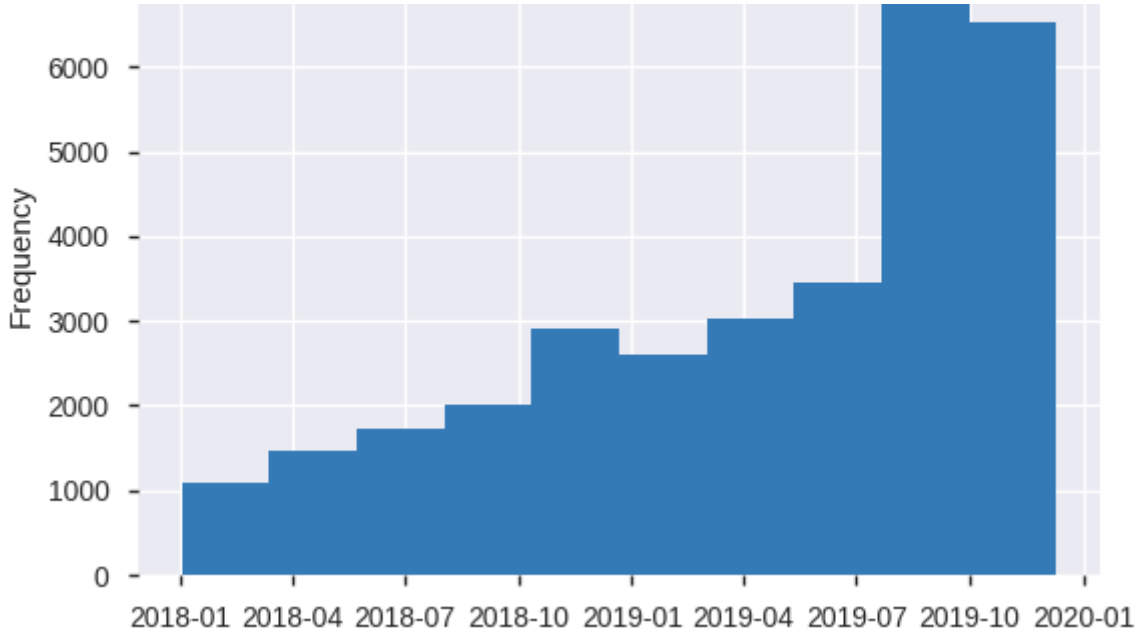
Maximum 5 values

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

DAT_ULTIMA_EVOLUCAO
Date

Distinct count	10439
Unique (%)	32.6%
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	2.1951
Minimum	0
Maximum	6
Zeros (%)	21.4%

Toggle details

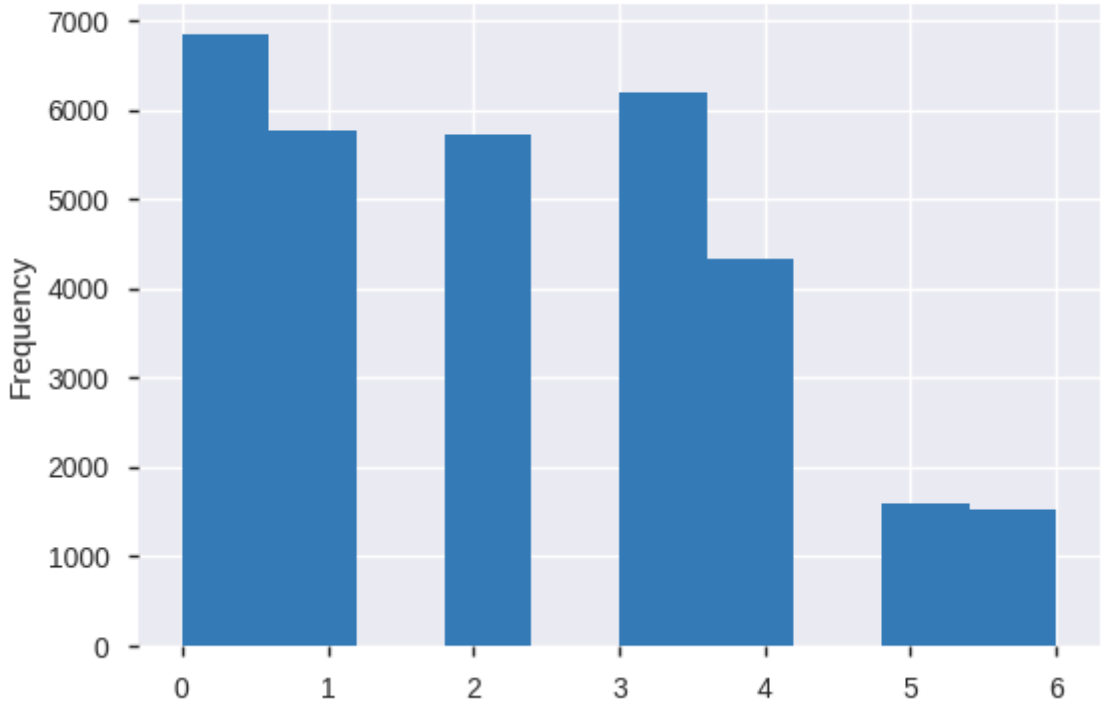
- Statistics
- Histogram
- Common Values
- Extreme Values

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	1.7147
Coef of variation	0.78114
Kurtosis	-0.7007
Mean	2.1951
MAD	1.4407
Skewness	0.4037
Sum	70270
Variance	2.9402
Memory size	250.2 KiB



Value	Count	Frequency (%)
0	6848	21.4%
3	6201	19.4%
1	5781	18.1%
2	5734	17.9%
4	4342	13.6%
5	1586	5.0%
6	1520	4.7%

Minimum 5 values

Value	Count	Frequency (%)
0	6848	21.4%
1	5781	18.1%
2	5734	17.9%
3	6201	19.4%
4	4342	13.6%

Maximum 5 values

Value	Count	Frequency (%)
2	5734	17.9%

3	6201	19.4%
4	4342	13.6%
5	1586	5.0%
6	1520	4.7%

PROTOCOLO
Numeric

Distinct count	402
Unique (%)	1.3%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	0.20669
Minimum	0
Maximum	1.2
Zeros (%)	0.4%

Toggle details

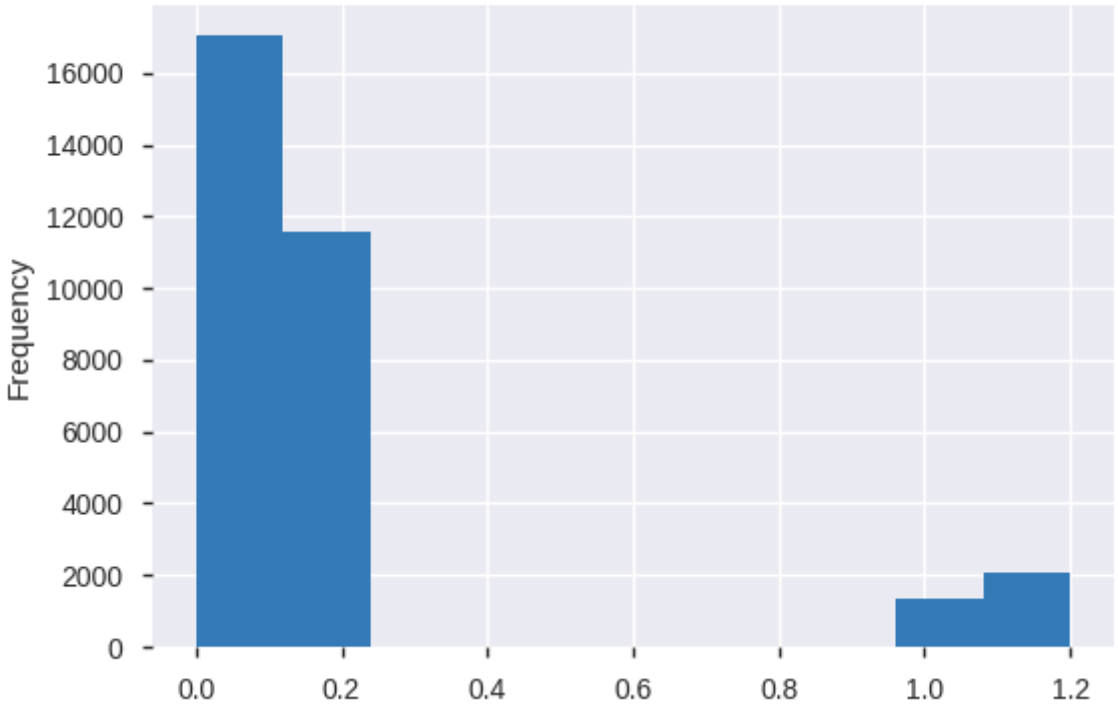
- Statistics
- Histogram
- Common Values
- Extreme Values

Quantile statistics

Minimum	0
5-th percentile	0.011
Q1	0.056
Median	0.112
Q3	0.168
95-th percentile	1.107
Maximum	1.2
Range	1.2
Interquartile range	0.112

Descriptive statistics

Standard deviation	0.31387
Coef of variation	1.5186
Kurtosis	4.2207
Mean	0.20669
MAD	0.19017
Skewness	2.4247
Sum	6616.5
Variance	0.098516
Memory size	250.2 KiB



Value	Count	Frequency (%)
0.006	178	0.6%
0.179	176	0.5%
0.16	173	0.5%
0.105	173	0.5%
0.104	171	0.5%
0.147	167	0.5%
0.196	164	0.5%
0.151	164	0.5%
0.155	163	0.5%
0.055	163	0.5%
Other values (392)	30320	94.7%

Minimum 5 values

Value	Count	Frequency (%)
0.0	137	0.4%
0.001	145	0.5%
0.002	135	0.4%
0.003	140	0.4%
0.004	123	0.4%

Maximum 5 values

Value	Count	Frequency (%)
1.196	23	0.1%
1.197	24	0.1%
1.198	14	0.0%
1.199	20	0.1%
1.2	16	0.0%

DURACAO
Categorical

Distinct count	13369
Unique (%)	41.8%
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 (13366)	31767

Toggle details

Value	Count	Frequency (%)
637 days 02:02:00	108	0.3%
1281 days 06:00:00	98	0.3%
1393 days 03:51:00	39	0.1%
393 days 02:59:00	36	0.1%
1347 days 02:03:00	29	0.1%
1632 days 00:10:00	27	0.1%
970 days 21:57:00	20	0.1%
721 days 04:00:00	17	0.1%
1191 days 00:56:00	17	0.1%
1630 days 01:00:00	16	0.0%
Other values (13359)	31605	98.7%

NDURACAO
Numeric

Distinct count	13369
Unique (%)	41.8%
Missing (%)	0.0%
Missing (n)	0
Infinite (%)	0.0%
Infinite (n)	0
Mean	927.11
Minimum	-333
Maximum	2349.2
Zeros (%)	0.0%

Toggle details

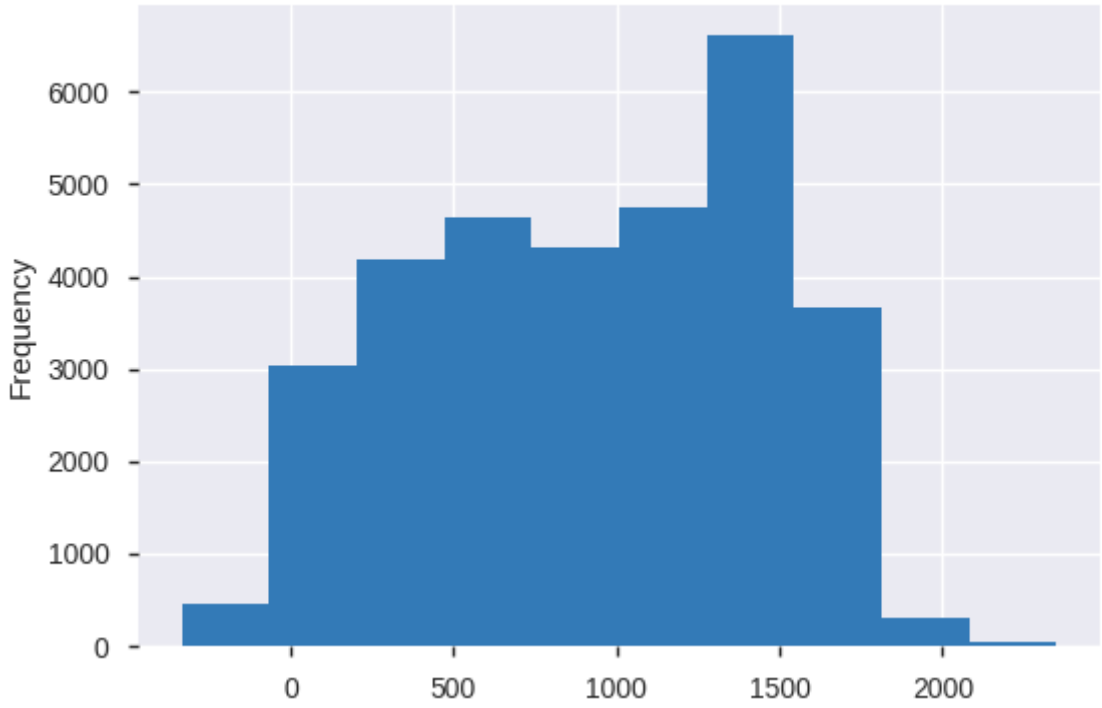
- Statistics
- Histogram
- Common Values
- Extreme Values

Quantile statistics

Minimum	-333
5-th percentile	73.128
Q1	490.86
Median	967.2
Q3	1397
95-th percentile	1634.8
Maximum	2349.2
Range	2682.2
Interquartile range	906.14

Descriptive statistics

Standard deviation	520.14
Coef of variation	0.56104
Kurtosis	-1.1015
Mean	927.11
MAD	452.16
Skewness	-0.18853
Sum	29679000
Variance	270550
Memory size	250.2 KiB



Value	Count	Frequency (%)
637.0847222222222	108	0.3%
1281.25	98	0.3%
1393.1604166666666	39	0.1%
393.12430555555557	36	0.1%

1347.0854166666666	29	0.1%
1632.0069444444443	27	0.1%
970.9145833333333	20	0.1%
721.1666666666666	17	0.1%
1191.0388888888888	17	0.1%
428.83194444444445	16	0.0%
Other values (13359)	31605	98.7%

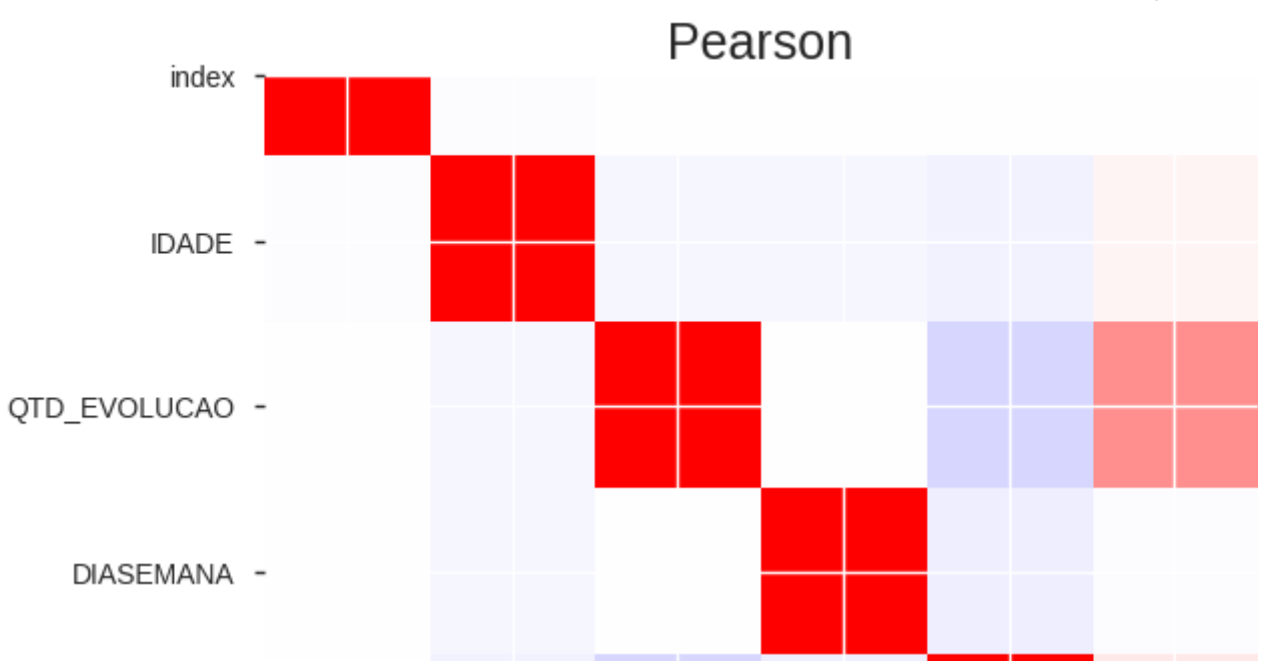
Minimum 5 values

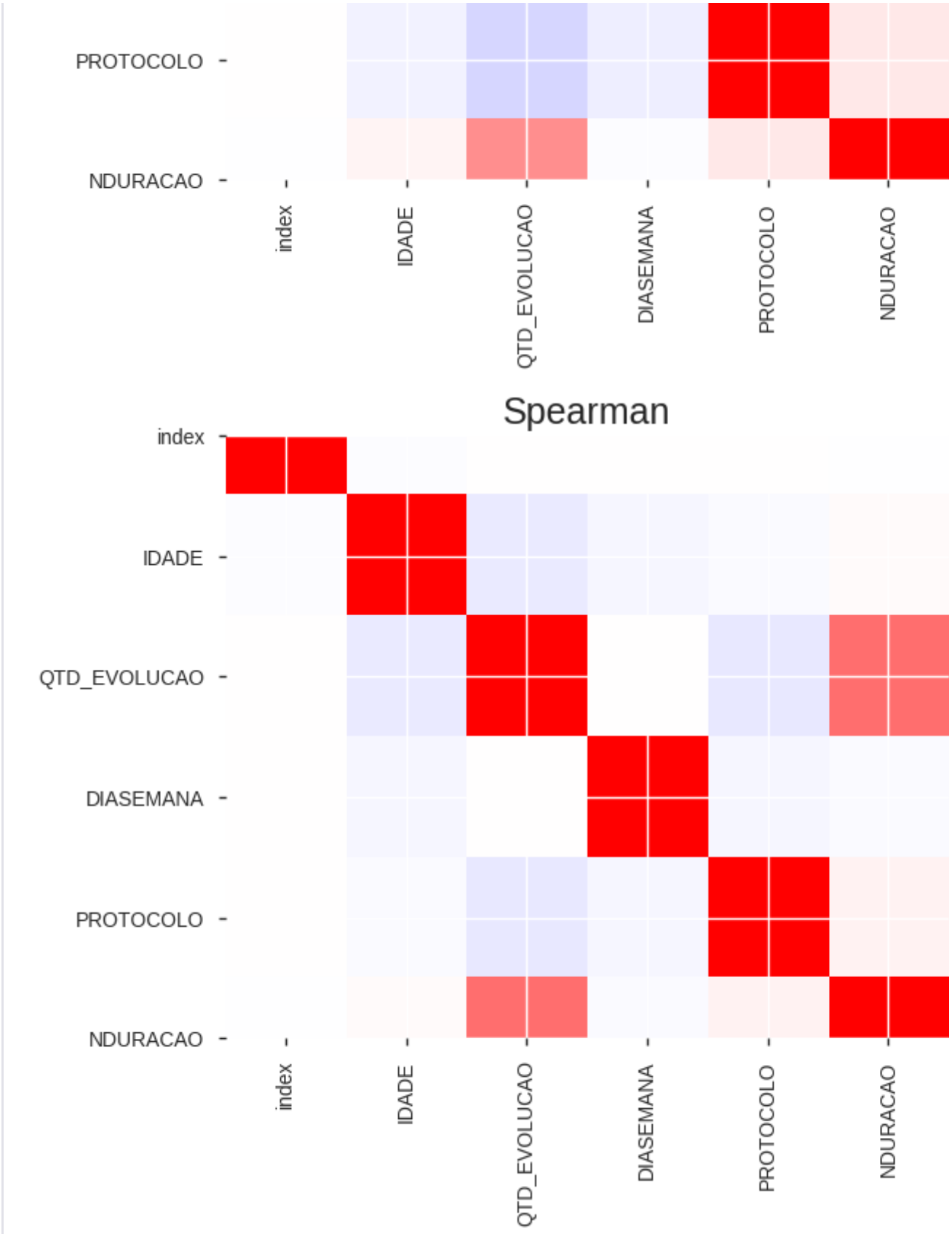
Value	Count	Frequency (%)
-333.0	1	0.0%
-301.80347222222224	1	0.0%
-286.83194444444445	1	0.0%
-286.8229166666667	1	0.0%
-279.87361111111113	2	0.0%

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_ATENDIME
```

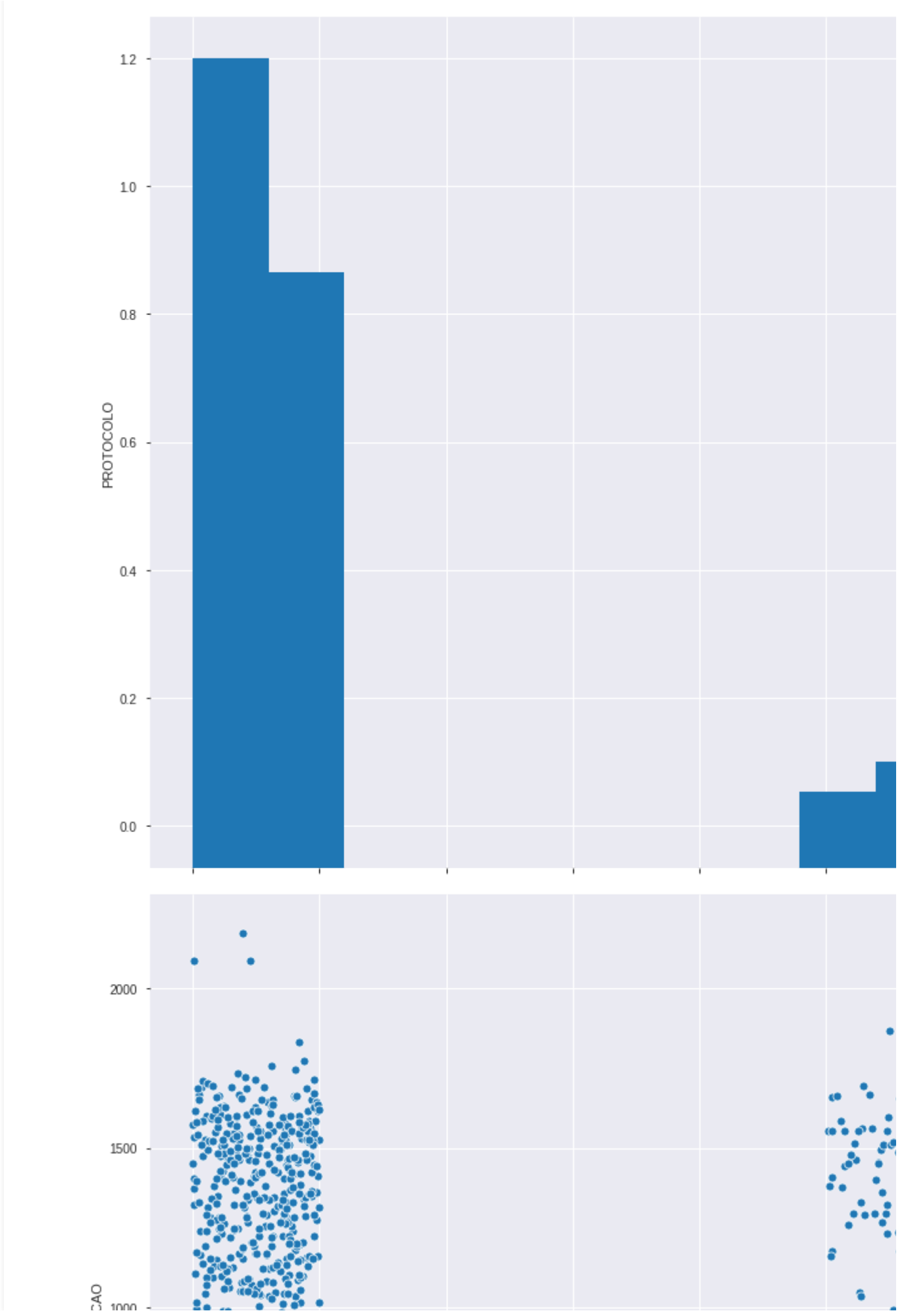
1	2018-12-04 10:00:00	RETORNO	AMBULATOI
3	2018-10-22 07:12:00	RETORNO	AMBULATOI
4	2018-09-20 13:00:00	RETORNO	AMBULATOI
5	2018-04-06 15:00:00	RETORNO	AMBULATOI

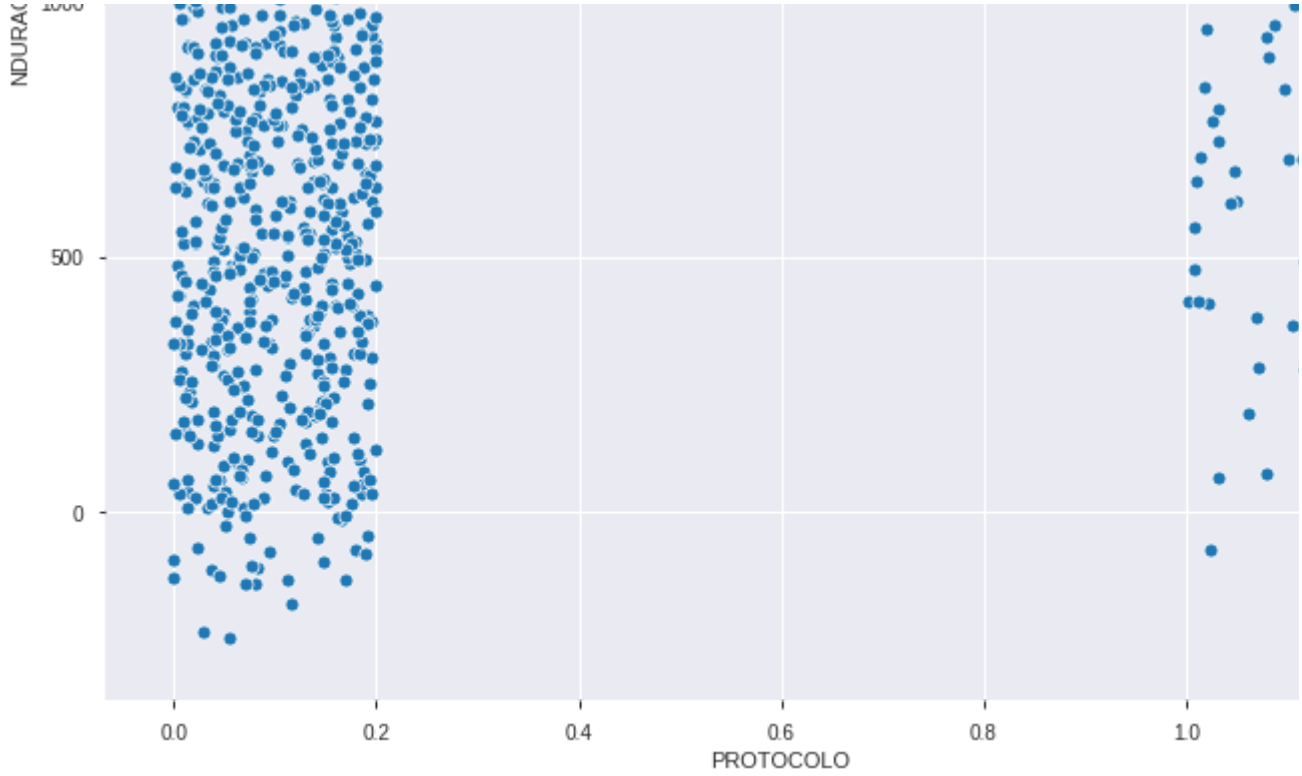
Análises

Plot do dataset puro

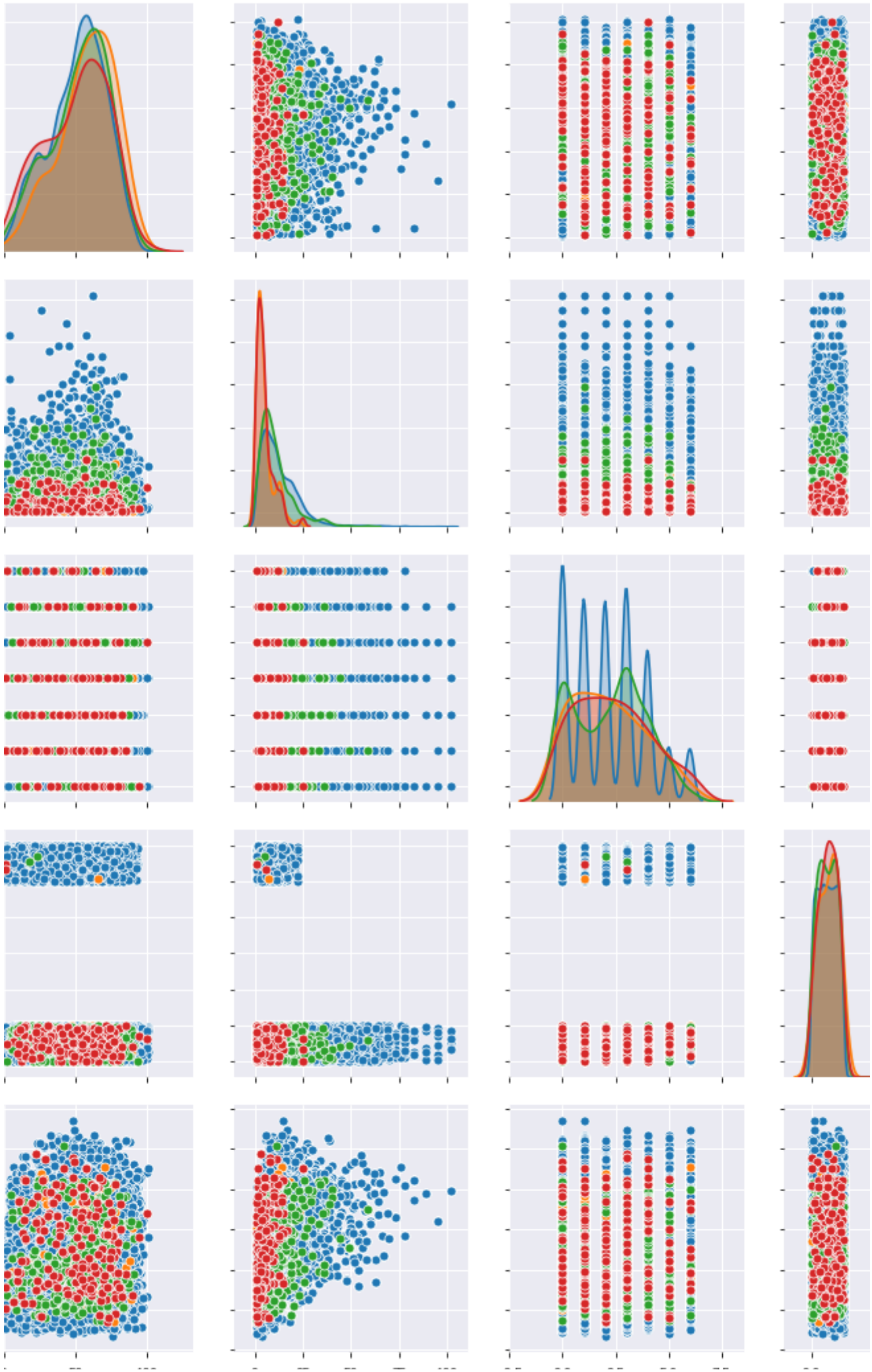
Protocolo x Duração

```
dfProtocoloDuracao = dfLimpo[['PROTOCOLO','NDURACAO']].sample(1000)
%matplotlib inline
sb.pairplot(dfProtocoloDuracao,height=10)
pl.show()
```


1.2
1.0
0.8
0.6
0.4
0.2
0.0
PROTOCOLO
2000
1500
1000




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




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

Clusterização

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

```
kmeans = KMeans(n_clusters=4, random_state=0)
```

```
dfCluster
```

	IDADE	PROTOCOLO	NDURACAO
1	13.039887	0.031	763.104167
3	51.941257	0.009	1233.008333
4	37.437147	0.108	515.893750
5	37.437147	0.069	515.893750
6	37.437147	0.191	515.893750
...	...	...	...
49178	74.020709	1.171	1729.039583
49179	60.056325	0.024	1335.247222
49182	77.390572	0.128	522.086806
49183	67.998791	1.033	708.041667
49186	58.470024	0.110	1519.061806

32012 rows × 3 columns

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

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=4, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=0, tol=0.0001, verbose=0)
```

```
kmeans.labels_
```

```
array([2, 0, 2, ..., 2, 2, 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
```

Regressão

```
<seaborn.axisgrid.PairGrid at 0x7f2186986590>
```

Double-click (or enter) to edit

```
dfLimpo
```

	DAT_HORA_ATENDIMENTO	NOM_ENCAMINHAMENTO	NOM_MODALIDADE_ATENDIMENTO	NOM_MUN
1	2018-12-04 10:00:00	RETORNO	AMBULATORIO	NOVO J
3	2018-10-22 07:12:00	RETORNO	AMBULATORIO	
4	2018-09-20 13:00:00	RETORNO	AMBULATORIO	IMPEI
5	2018-04-06 15:00:00	RETORNO	AMBULATORIO	IMPEI
6	2018-09-04 12:33:00	RETORNO	AMBULATORIO	IMPEI
...	...	...	...	
49178	2018-07-05 07:00:00	RETORNO	AMBULATORIO	OI
49179	2018-04-10 12:00:00	RETORNO	AMBULATORIO	CRI
49182	2018-05-04 12:08:00	RETORNO	AMBULATORIO	PARA(PA
49183	2018-02-19 07:10:00	RETORNO	AMBULATORIO	OI
49186	2018-08-29 12:33:00	RETORNO	AMBULATORIO	M

32012 rows × 18 columns

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

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

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

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

dfRegressao
```

	NDURACAO	PROTOCOLO	NOM_MODALIDADE_ATENDIMENTO	AMBULATORIO	INTERNAÇÃO	E
1	763.104167	0.031	AMBULATORIO	1	0	
3	1233.008333	0.009	AMBULATORIO	1	0	
4	515.893750	0.108	AMBULATORIO	1	0	
5	515.893750	0.069	AMBULATORIO	1	0	
6	515.893750	0.191	AMBULATORIO	1	0	
...	...	...	...	...	...	
49178	1729.039583	1.171	AMBULATORIO	1	0	
49179	1335.247222	0.024	AMBULATORIO	1	0	
49182	522.086806	0.128	AMBULATORIO	1	0	
49183	708.041667	1.033	AMBULATORIO	1	0	
49186	1519.061806	0.110	AMBULATORIO	1	0	

```
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

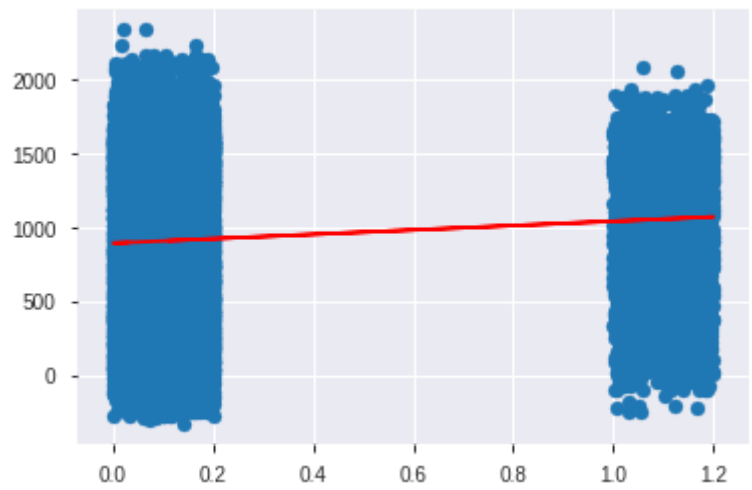
```
teste = [[0,1,0,0,0]]
```

```
model.predict(teste)
```

```
array([[902.79729771]])
```

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

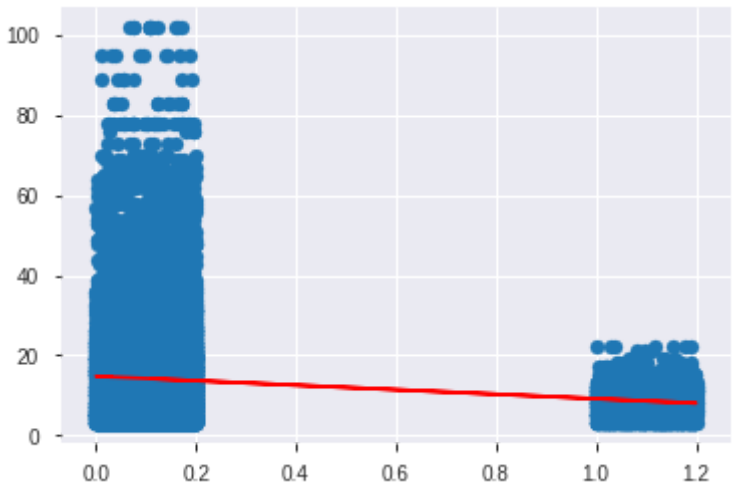
Double-click (or enter) to edit

```
%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
```

dfRegressaoPlot

	NDURACAO	PROTOCOLO	NOM_MODALIDADE_ATENDIMENTO	QTD_EVOLUCAO	AMBULATORIO
1	763.104167	0.031	AMBULATORIO	14	1
3	1233.008333	0.009	AMBULATORIO	9	1
4	515.893750	0.108	AMBULATORIO	13	1
5	515.893750	0.069	AMBULATORIO	13	1
6	515.893750	0.191	AMBULATORIO	13	1
...	...	...	...	...	...
49176	583.083333	0.171	AMBULATORIO	6	1
49178	1729.039583	1.171	AMBULATORIO	8	1
49179	1335.247222	0.024	AMBULATORIO	16	1
49183	708.041667	1.033	AMBULATORIO	6	1
49186	1519.061806	0.110	AMBULATORIO	17	1

```
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()
```



Conclusão

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