

▼ 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 plt
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: 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: packaging>=16.8 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: numpy>=1.7.1 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: olefile in /usr/local/lib/python3.6/dist-packages (frc
Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.6/dist-pack
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_PF
```

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'] <= '2018-12-31')]
```

```
#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_MODALIDADE_ATENDIMENTO"].value_counts()
```

```

count
AMBULATORIO          31361
INTERNAÇÃO             397
SADT EXTERNO           187
SADT UBS MARILIA        67
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['DURACAO'] = dfLimpo['DAT_ULTIMA_EVOLUCAO'].sub(dfLimpo['DAT_HORA_ANAMNESE'], axis=1)
dfLimpo['DURACAO'] = dfLimpo['DURACAO'] / np.timedelta64(1, 'D')

```

```
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
```



```
try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/usage>

```
# This is added back by InteractiveShellApp.init_path()
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:12: SettingWithCopyWarning:
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/usage>

```
if sys.path[0] == '':
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:13: SettingWithCopyWarning:
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/usage>

```
del sys.path[0]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:15: SettingWithCopyWarning:
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/usage>

```
from ipykernel import kernelapp as app
```

	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	BRL
212	2018-08-06 12:02:00	RETORNO	SADT EXTERNO	MORTI
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	↑ SAN
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	LI

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	BRL
212	2018-08-06 12:02:00	RETORNO	SADT EXTERNO	MORTI
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	↑ SAN
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	LI

155 rows × 18 columns

▼ Profiling

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



1547.0634100000000	23	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%

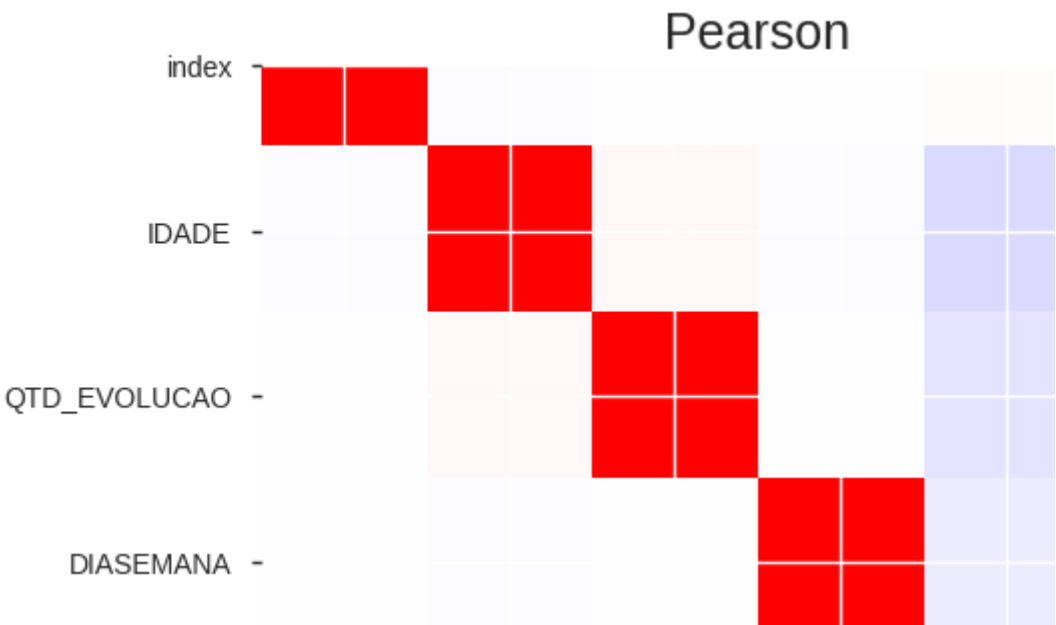
Minimum 5 values

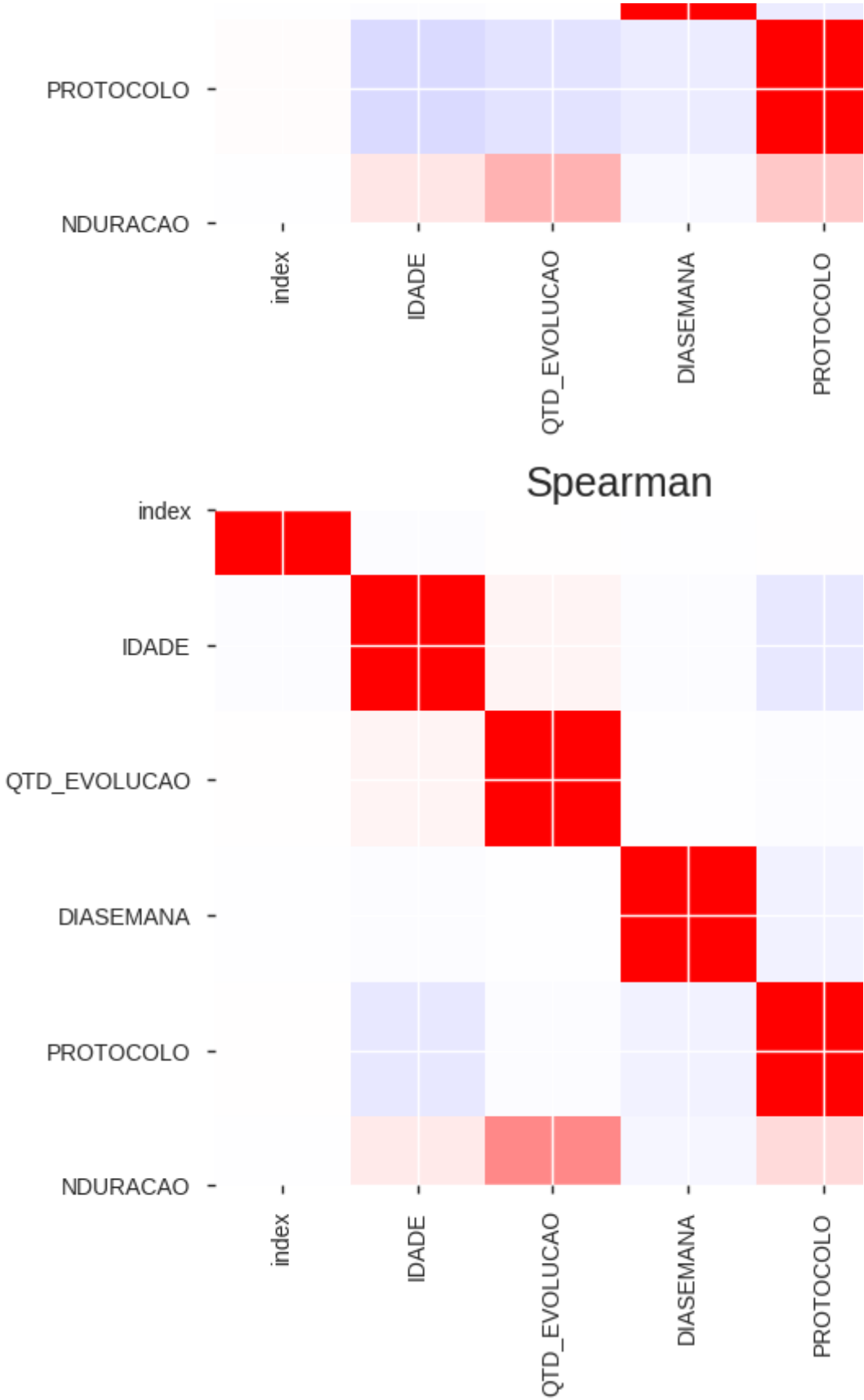
Value	Count	Frequency (%)
-333.0	1	0.0%
-301.80347222222224	1	0.0%
-286.83194444444445	1	0.0%
-286.8229166666667	1	0.0%
-279.87361111111113	2	0.0%

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	NOM
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	
14	2018-11-19 07:00:00	RETORNO	AMBULATORIO	



