UNIFOR - Universidade de Fortaleza

DISCIPLINA: Machine Learning

PROFESSOR: Carlos Caminha

TURMA W001: MBA em Ciência de Dados

EQUIPE 01: Diego Teixeira Marques - 1924526; Mario Gibson Maia Pinto - 1924979; Marcos Godim - 1924529; Jarison James Lima de Melo - 1924527

Objetivos: O presente trabalho visa desenvolver um experimento utilizando os algorítmos de clusterziração KMEANS e DBScan mostrando os resultados obtidos sob uma base de dados da PRF que contém os registros de acidentes nas rodovias federais do Brasil no ano de 2019. Observa-se, como não há participação de especialistas na área inerente aos dados trabalhados neste estudo, os resultados não possuem necessariamente uma aplicação prática como resolução de problemas relacionados aos acidentes nas rodovias brasileiras.

Resumo: O trabalho foi desenvolvido com algumas premissas de cunho experimental, possibilitando o acompanhamento de forma estruturada do fluxo de tratamento de features até os resultados obtidos pós clusterização. Em suma, as estapas são dividivas em importação de bibliotecas, carga de dados, análise e tratamento de dados, visualização dos dados, execução de algorítmos e visualização de resultados.

Link - Base de Dados: https://arquivos.prf.gov.br/arquivos/index.php/s/sdvJndbl5wLyh3J (https://arquivos.prf.gov.br/arquivos/index.php/s/sdvJndbl5wLyh3J)

Link - Dicionário de Dados: https://portal.prf.gov.br/dados-abetos-dicionario-acidentes (https://portal.prf.gov.br/dados-abetos-dicionario-acidentes)

Modelo de ML: Cluster

Features utilizadas = ['uf', 'causa_principal', 'causa_acidente', 'tipo_acidente', 'classificacao_acidente', 'fase_dia', 'sentido_via', 'condicao_metereologica', 'tipo_pista', 'tracado_via', 'uso_solo', 'tipo_veiculo', 'tipo_envolvido', 'estado_fisico', 'sexo']

Target = Não especificado.

Importando Bibliotecas

In [55]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import OneHotEncoder, LabelEncoder, MinMaxScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.cluster import DBSCAN
import matplotlib.cm as cm
%matplotlib inline
```

Carregando os Dados

In [56]:
```
df_dados = pd.read_csv('acidentes2019_todas_causas_tipos.csv',  sep=';', encodin
df_dados
```

Out[56]:

	id	pesid	data_inversa	dia_semana	horario	uf	br	km	municipio	c
0	182210.0	402103.0	2019-01-01	terça-feira	01:30:00	SP	116.0	218	GUARULHOS	
1	182210.0	402106.0	2019-01-01	terça-feira	01:30:00	SP	116.0	218	GUARULHOS	
2	182210.0	402104.0	2019-01-01	terça-feira	01:30:00	SP	116.0	218	GUARULHOS	
3	182210.0	402102.0	2019-01-01	terça-feira	01:30:00	SP	116.0	218	GUARULHOS	
4	182211.0	402126.0	2019-01-01	terça-feira	01:30:00	PR	373.0	177,3	PONTA GROSSA	
...	...	...	...	...	...	...	...	...	...	
331661	266434.0	593948.0	2019-06-03	segunda-feira	19:00:00	CE	222.0	1,4	CAUCAIA	
331662	266434.0	593947.0	2019-06-03	segunda-feira	19:00:00	CE	222.0	1,4	CAUCAIA	
331663	266573.0	594270.0	2019-07-13	sábado	19:35:00	PR	373.0	425	CANDOI	
331664	266573.0	594270.0	2019-07-13	sábado	19:35:00	PR	373.0	425	CANDOI	
331665	266627.0	594393.0	2019-10-22	terça-feira	20:20:00	BA	324.0	608	SIMOES FILHO	

331666 rows × 37 columns

Analisando e Tratando os Dados

Fase 1

In [57]:
```
# Analisando volumetria do DataFrame

df_dados.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 331666 entries, 0 to 331665
Data columns (total 37 columns):
id                          331666 non-null float64
pesid                       307223 non-null float64
data_inversa                331666 non-null object
dia_semana                  331666 non-null object
horario                     331666 non-null object
uf                          331666 non-null object
br                          331291 non-null float64
km                          331291 non-null object
municipio                   331666 non-null object
causa_principal             331666 non-null object
causa_acidente              331666 non-null object
ordem_tipo_acidente         331626 non-null float64
tipo_acidente               331626 non-null object
classificacao_acidente      331666 non-null object
fase_dia                    331666 non-null object
sentido_via                 331666 non-null object
condicao_metereologica      331666 non-null object
tipo_pista                  331666 non-null object
tracado_via                 331666 non-null object
uso_solo                    331666 non-null object
id_veiculo                  331666 non-null int64
tipo_veiculo                331666 non-null object
marca                       317602 non-null object
ano_fabricacao_veiculo      314393 non-null float64
tipo_envolvido              331666 non-null object
estado_fisico               331666 non-null object
idade                       269798 non-null float64
sexo                        331666 non-null object
ilesos                      331666 non-null int64
feridos_leves               331666 non-null int64
feridos_graves              331666 non-null int64
mortos                      331666 non-null int64
latitude                    331666 non-null object
longitude                   331666 non-null object
regional                    331666 non-null object
delegacia                   331666 non-null object
uop                         314468 non-null object
dtypes: float64(6), int64(5), object(26)
memory usage: 60.7+ MB
```

O dataframe possui 37 colunas. Observa-se 26 features do tipo object com grande potencial de ser um dado categórico. Vamos analisar!

In [58]:
```
# Selecionando apenas features do tipo object

df_object = df_dados.select_dtypes(include='object')
df_object
```

Out[58]:

	data_inversa	dia_semana	horario	uf	km	municipio	causa_principal	causa_ac
0	2019-01-01	terça-feira	01:30:00	SP	218	GUARULHOS	Sim	F Ate Cor
1	2019-01-01	terça-feira	01:30:00	SP	218	GUARULHOS	Sim	F Ate Cor
2	2019-01-01	terça-feira	01:30:00	SP	218	GUARULHOS	Sim	F Ate Cor
3	2019-01-01	terça-feira	01:30:00	SP	218	GUARULHOS	Sim	F Ate Cor
4	2019-01-01	terça-feira	01:30:00	PR	177,3	PONTA GROSSA	Sim	F Ate Cor
...	...	...	...	...	...	...	...	
331661	2019-06-03	segunda-feira	19:00:00	CE	1,4	CAUCAIA	Sim	Não g distâr seg
331662	2019-06-03	segunda-feira	19:00:00	CE	1,4	CAUCAIA	Sim	Não g distâr seg
331663	2019-07-13	sábado	19:35:00	PR	425	CANDOI	Sim	I Mecâr \
331664	2019-07-13	sábado	19:35:00	PR	425	CANDOI	Sim	I Mecâr \
331665	2019-10-22	terça-feira	20:20:00	BA	608	SIMOES FILHO	Sim	Defeito

331666 rows × 26 columns

In [59]:
```
df_object.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 331666 entries, 0 to 331665
Data columns (total 26 columns):
data_inversa              331666 non-null object
dia_semana                331666 non-null object
horario                   331666 non-null object
uf                        331666 non-null object
km                        331291 non-null object
municipio                 331666 non-null object
causa_principal           331666 non-null object
causa_acidente            331666 non-null object
tipo_acidente             331626 non-null object
classificacao_acidente    331666 non-null object
fase_dia                  331666 non-null object
sentido_via               331666 non-null object
condicao_metereologica    331666 non-null object
tipo_pista                331666 non-null object
tracado_via               331666 non-null object
uso_solo                  331666 non-null object
tipo_veiculo              331666 non-null object
marca                     317602 non-null object
tipo_envolvido            331666 non-null object
estado_fisico             331666 non-null object
sexo                      331666 non-null object
latitude                  331666 non-null object
longitude                 331666 non-null object
regional                  331666 non-null object
delegacia                 331666 non-null object
uop                       314468 non-null object
dtypes: object(26)
memory usage: 32.9+ MB
```

Fase 2

Vamos retirar os dados referentes ao tempo e que indicam a corporação que atendeu a ocorrência.

In [60]:
```
# Reduzindo colunas

df_object.drop(['data_inversa', 'dia_semana', 'horario', 'regional', 'delegacia'
```

```
c:\users\diego\appdata\local\programs\python\python37-32\lib\site-packages\pand
as\core\frame.py:4102: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stab
le/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydat
a.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y)
  errors=errors,
```

In [61]:
```
# Análise geral do Dataframe reduzido

df_object.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 331666 entries, 0 to 331665
Data columns (total 20 columns):
uf                          331666 non-null object
km                          331291 non-null object
municipio                   331666 non-null object
causa_principal             331666 non-null object
causa_acidente              331666 non-null object
tipo_acidente               331626 non-null object
classificacao_acidente      331666 non-null object
fase_dia                    331666 non-null object
sentido_via                 331666 non-null object
condicao_metereologica      331666 non-null object
tipo_pista                  331666 non-null object
tracado_via                 331666 non-null object
uso_solo                    331666 non-null object
tipo_veiculo                331666 non-null object
marca                       317602 non-null object
tipo_envolvido              331666 non-null object
estado_fisico               331666 non-null object
sexo                        331666 non-null object
latitude                    331666 non-null object
longitude                   331666 non-null object
dtypes: object(20)
memory usage: 25.3+ MB
```

In [62]:
```
# Verificando valores nulo

df_object.isnull().sum()
```

Out[62]:
```
uf                            0
km                          375
municipio                     0
causa_principal               0
causa_acidente                0
tipo_acidente                40
classificacao_acidente        0
fase_dia                      0
sentido_via                   0
condicao_metereologica        0
tipo_pista                    0
tracado_via                   0
uso_solo                      0
tipo_veiculo                  0
marca                     14064
tipo_envolvido                0
estado_fisico                 0
sexo                          0
latitude                      0
longitude                     0
dtype: int64
```

Interressante analisarmos as 3 colunas que apresentam valores nulos...

In [63]:
```
print(df_object.km.value_counts(), '\n\n\n      *****        \n\n')
print(df_object.tipo_acidente.value_counts(), '\n\n\n      *****        \n\n')
print(df_object.marca.value_counts(), '\n\n\n      *****        \n\n')
```

```
1        1318
3        1286
4         989
2         944
5         892
         ...
560,5       1
905,9       1
419,4       1
937,8       1
799,1       1
Name: km, Length: 7918, dtype: int64


      *****


Colisão traseira                  59084
Saída de leito carroçável         43242
Colisão lateral                   36249
Colisão transversal               32835
Tombamento                        31960
Colisão frontal                   26398
Colisão com objeto estático       25944
Queda de ocupante de veículo      22886
Capotamento                       20585
Atropelamento de Pedestre         12590
Engavetamento                      7875
Incêndio                           3357
Derramamento de carga              3121
Atropelamento de Animal            2734
Colisão com objeto em movimento    1813
Danos eventuais                     953
Name: tipo_acidente, dtype: int64


      *****


SR/RANDON SR CA             4282
MBENZ/MPOLO PARADISO R      3686
VW/GOL 1.0                  3533
HONDA/CG 125 FAN KS         3133
HONDA/CG 150 TITAN KS       2758
                            ...
SCANIA/BUSSCAR PANORAM R       1
IVECO/WAYCLASS 70C17HDE        1
I/JEEP GCHEROKEE LTD 5.7       1
RENAULT/MEGANESD PRI 20A       1
JTA/SUZUKI BANDIT N1200        1
Name: marca, Length: 6308, dtype: int64
```

*****

Fase 3

Vamos analisar a quantidade de valores únicos em cada feature. Interessante evitar as que possuem muita variedade nesta primeira versão.

In [64]:
```
# Verificando quantidade de categorias por feature

for i in df_object.columns:
  print(i, ' :', len(df_object[str(i)].unique()))
```

```
uf  : 27
km  : 7919
municipio  : 1767
causa_principal  : 2
causa_acidente  : 24
tipo_acidente  : 17
classificacao_acidente  : 3
fase_dia  : 4
sentido_via  : 3
condicao_metereologica  : 10
tipo_pista  : 3
tracado_via  : 10
uso_solo  : 2
tipo_veiculo  : 24
marca  : 6309
tipo_envolvido  : 6
estado_fisico  : 5
sexo  : 4
latitude  : 37325
longitude  : 37329
```

In [65]:
```
# Reduzindo colunas

df_object.drop(['km', 'municipio', 'marca', 'latitude', 'longitude'], axis=1, in
df_object.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 331666 entries, 0 to 331665
Data columns (total 15 columns):
uf                          331666 non-null object
causa_principal             331666 non-null object
causa_acidente              331666 non-null object
tipo_acidente               331626 non-null object
classificacao_acidente      331666 non-null object
fase_dia                    331666 non-null object
sentido_via                 331666 non-null object
condicao_metereologica      331666 non-null object
tipo_pista                  331666 non-null object
tracado_via                 331666 non-null object
uso_solo                    331666 non-null object
tipo_veiculo                331666 non-null object
tipo_envolvido              331666 non-null object
estado_fisico               331666 non-null object
sexo                        331666 non-null object
dtypes: object(15)
memory usage: 19.0+ MB
```

In [66]:
```
# Verificando a quantidade de nulos

df_object.isnull().sum()
```

Out[66]:
```
uf                          0
causa_principal             0
causa_acidente              0
tipo_acidente              40
classificacao_acidente      0
fase_dia                    0
sentido_via                 0
condicao_metereologica      0
tipo_pista                  0
tracado_via                 0
uso_solo                    0
tipo_veiculo                0
tipo_envolvido              0
estado_fisico               0
sexo                        0
dtype: int64
```

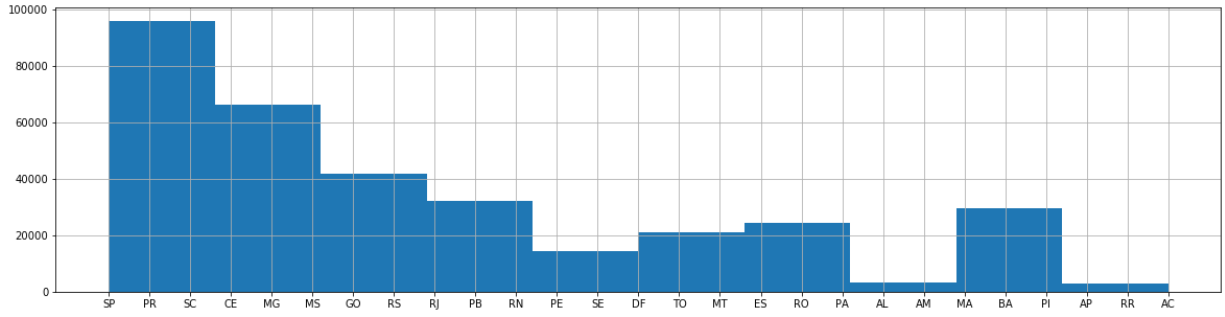
In [67]:
```
# Verificando quantidade de categorias por feature

for i in df_object.columns:
  print(i, ' :', len(df_object[str(i)].unique()))
```

```
uf  : 27
causa_principal  : 2
causa_acidente  : 24
tipo_acidente  : 17
classificacao_acidente  : 3
fase_dia  : 4
sentido_via  : 3
condicao_metereologica  : 10
tipo_pista  : 3
tracado_via  : 10
uso_solo  : 2
tipo_veiculo  : 24
tipo_envolvido  : 6
estado_fisico  : 5
sexo  : 4
```

Vamos analisar as UFs mais recorrentes neste DataFrame.

In [68]:
```
# Histograma das UFs: SP, PR e SC são os estados que mais tiveram registros de ac

df_object.uf.hist(bins=10, figsize=(20,5))
```

Out[68]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24affdf0>
```



Fase 4

In [69]:
```
# Visualizando o Dataframe para esta fase

df_object
```

Out[69]:

	uf	causa_principal	causa_acidente	tipo_acidente	classificacao_acidente	fase_dia	senti
0	SP	Sim	Falta de Atenção à Condução	Colisão com objeto estático	Com Vítimas Feridas	Plena Noite	Cre
1	SP	Sim	Falta de Atenção à Condução	Colisão com objeto estático	Com Vítimas Feridas	Plena Noite	Cre
2	SP	Sim	Falta de Atenção à Condução	Colisão com objeto estático	Com Vítimas Feridas	Plena Noite	Cre
3	SP	Sim	Falta de Atenção à Condução	Colisão com objeto estático	Com Vítimas Feridas	Plena Noite	Cre
4	PR	Sim	Falta de Atenção à Condução	Colisão traseira	Com Vítimas Feridas	Plena Noite	Cre
...	...	...	...	...	...	...	
331661	CE	Sim	Não guardar distância de segurança	Colisão traseira	Com Vítimas Feridas	Plena Noite	Cre
331662	CE	Sim	Não guardar distância de segurança	Colisão traseira	Com Vítimas Feridas	Plena Noite	Cre
331663	PR	Sim	Defeito Mecânico no Veículo	Capotamento	Com Vítimas Feridas	Plena Noite	Decre
331664	PR	Sim	Defeito Mecânico no Veículo	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331665	BA	Sim	Defeito na Via	Queda de ocupante de veículo	Com Vítimas Feridas	Plena Noite	Cre

331666 rows × 15 columns

Olhando a qualidade dos dados e fazendo as transformações julgadas como necessárias.

In [70]:
```
df_object.isnull().sum()
```

Out[70]:
```
uf                         0
causa_principal            0
causa_acidente             0
tipo_acidente             40
classificacao_acidente     0
fase_dia                   0
sentido_via                0
condicao_metereologica     0
tipo_pista                 0
tracado_via                0
uso_solo                   0
tipo_veiculo               0
tipo_envolvido             0
estado_fisico              0
sexo                       0
dtype: int64
```

In [71]:
```
# Listando registros com tipo_acidente nulo

df_object[df_object.isnull().tipo_acidente == True]
```

Out[71]:

	uf	causa_principal	causa_acidente	tipo_acidente	classificacao_acidente	fase_dia	s
3969	SC	Sim	Defeito Mecânico no Veículo	NaN	Sem Vítimas	Pleno dia	
3970	SC	Sim	Defeito Mecânico no Veículo	NaN	Sem Vítimas	Pleno dia	
3971	SC	Sim	Defeito Mecânico no Veículo	NaN	Sem Vítimas	Pleno dia	
5910	RS	Sim	Falta de Atenção à Condução	NaN	Com Vítimas Feridas	Plena Noite	D
5911	RS	Sim	Falta de Atenção à Condução	NaN	Com Vítimas Feridas	Plena Noite	D

In [72]:
```
# Pegando index das linhas para visualizar o resultado da transformação

index = df_object[df_object.isnull().tipo_acidente == True].index
index
```

Out[72]:
```
Int64Index([ 3969,  3970,  3971,  5910,  5911,  9942,  9943, 11626, 11627,
            11628, 11629, 11630, 11631, 11632, 11633, 12183, 12184, 12185,
            12186, 15590, 15591, 15642, 17778, 17779, 20202, 24275, 26252,
            31199, 31200, 31201, 32804, 32805, 36029, 36581, 36582, 36583,
            36584, 38216, 38217, 38904],
           dtype='int64')
```

In [73]:
```
# Substituindo valor e mostrando o resultado

df_object.tipo_acidente = df_object.tipo_acidente.fillna('Não Informado')
df_object.iloc[index]
```

```
c:\users\diego\appdata\local\programs\python\python37-32\lib\site-packages\pa
ndas\core\generic.py:5208: 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/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.p
ydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
-a-copy)
  self[name] = value
```

In [74]:
```
# Verificando se ainda há algum valor nulo

df_object.isnull().sum()
```

Out[74]:
```
uf                      0
causa_principal         0
causa_acidente          0
tipo_acidente           0
classificacao_acidente  0
fase_dia                0
sentido_via             0
condicao_metereologica  0
tipo_pista              0
tracado_via             0
uso_solo                0
tipo_veiculo            0
tipo_envolvido          0
estado_fisico           0
sexo                    0
dtype: int64
```

In [75]:
```
# Analisando volumetria por categoria

df_object.causa_principal.value_counts()
```

Out[75]:
```
Sim    259263
Não     72403
Name: causa_principal, dtype: int64
```

In [76]:
```
# Analisando volumetria por categoria

df_object.causa_acidente.value_counts()
```

Out[76]:
```
Falta de Atenção à Condução
109288
Desobediência às normas de trânsito pelo condutor
44594
Velocidade Incompatível
39619
Não guardar distância de segurança
25773
Ingestão de Álcool
23081
Defeito Mecânico no Veículo
13461
Pista Escorregadia
11434
Condutor Dormindo
11218
Ultrapassagem Indevida
9304
Falta de Atenção do Pedestre
6265
Restrição de Visibilidade
5527
Defeito na Via
5278
Animais na Pista
4835
Avarias e/ou desgaste excessivo no pneu
4352
Sinalização da via insuficiente ou inadequada
2795
Mal Súbito
2548
Objeto estático sobre o leito carroçável
2273
Fenômenos da Natureza
2149
Carga excessiva e/ou mal acondicionada
1602
Desobediência às normas de trânsito pelo pedestre
1539
Ingestão de álcool e/ou substâncias psicoativas pelo pedestre
1446
Deficiência ou não Acionamento do Sistema de Iluminação/Sinalização do Veículo
1389
Agressão Externa
1373
Ingestão de Substâncias Psicoativas
523
Name: causa_acidente, dtype: int64
```

In [77]:
```
df_object.tipo_acidente.value_counts()
```

Out[77]:
```
Colisão traseira                    59084
Saída de leito carroçável           43242
Colisão lateral                     36249
Colisão transversal                 32835
Tombamento                          31960
Colisão frontal                     26398
Colisão com objeto estático         25944
Queda de ocupante de veículo        22886
Capotamento                         20585
Atropelamento de Pedestre           12590
Engavetamento                        7875
Incêndio                             3357
Derramamento de carga                3121
Atropelamento de Animal              2734
Colisão com objeto em movimento      1813
Danos eventuais                       953
Não Informado                          40
Name: tipo_acidente, dtype: int64
```

In [78]:
```
# Analisando volumetria por categoria

df_object.classificacao_acidente.value_counts()
```

Out[78]:
```
Com Vítimas Feridas    248480
Com Vítimas Fatais      42509
Sem Vítimas             40677
Name: classificacao_acidente, dtype: int64
```

In [79]:
```
# Analisando volumetria por categoria

df_object.fase_dia.value_counts()
```

Out[79]:
```
Pleno dia      182200
Plena Noite    114176
Anoitecer       18741
Amanhecer       16549
Name: fase_dia, dtype: int64
```

In [80]:
```
# Analisando volumetria por categoria

df_object.sentido_via.value_counts()
```

Out[80]:
```
Crescente        177251
Decrescente      154040
Não Informado       375
Name: sentido_via, dtype: int64
```

In [81]:
```
# Analisando volumetria por categoria

df_object.condicao_metereologica.value_counts()
```

Out[81]:
```
Céu Claro           188447
Nublado              56584
Chuva                39470
Sol                  26334
Garoa/Chuvisco       12222
Nevoeiro/Neblina      4036
Ignorado              3969
Vento                  587
Granizo                 16
Neve                     1
Name: condicao_metereologica, dtype: int64
```

In [82]:
```
# Analisando volumetria por categoria

df_object.tipo_pista.value_counts()
```

Out[82]:
```
Simples     179008
Dupla       127712
Múltipla     24946
Name: tipo_pista, dtype: int64
```

In [83]:
```
# Analisando volumetria por categoria

df_object.tracado_via.value_counts()
```

Out[83]:
```
Reta                    197020
Curva                    50260
Não Informado            47083
Interseção de vias       13634
Desvio Temporário        10355
Rotatória                 5500
Retorno Regulamentado     3654
Ponte                     1992
Viaduto                   1819
Túnel                      349
Name: tracado_via, dtype: int64
```

In [84]:
```
# Analisando volumetria por categoria
# Segundo dicionario de dados: Sim: Urbano, Não: Rural

df_object.uso_solo.value_counts()
```

Out[84]:
```
Não    195177
Sim    136489
Name: uso_solo, dtype: int64
```

In [85]:
```
# Analisando volumetria por categoria

df_object.tipo_veiculo.value_counts()
```

Out[85]:
```
Automóvel            131872
Motocicleta           57211
Caminhonete           28134
Semireboque           24881
Caminhão-trator       23610
Caminhão              21801
Ônibus                14851
Camioneta              8055
Motoneta               6586
Utilitário             3898
Bicicleta              3793
Micro-ônibus           2753
Reboque                1934
Outros                  968
Ciclomotor              805
Carroça-charrete        158
Trator de rodas         152
Não Informado           120
Triciclo                 48
Trem-bonde               12
Trator misto             10
Quadriciclo               7
Carro de mão              6
Trator de esteira         1
Name: tipo_veiculo, dtype: int64
```

In [86]:
```
# Analisando volumetria por categoria

df_object.tipo_envolvido.value_counts()
```

Out[86]:
```
Condutor         199180
Passageiro        85504
Não Informado     24443
Testemunha        15257
Pedestre           7195
Cavaleiro            87
Name: tipo_envolvido, dtype: int64
```

In [87]:
```
# Analisando volumetria por categoria

df_object.estado_fisico.value_counts()
```

Out[87]:
```
Ileso            121795
Lesões Leves     110405
Não Informado     50420
Lesões Graves     36383
Óbito             12663
Name: estado_fisico, dtype: int64
```

In [88]:
```
# Analisando volumetria por categoria

df_object.sexo.value_counts()
```

Out[88]:
```
Masculino        209855
Feminino          69478
Não Informado     50420
Ignorado           1913
Name: sexo, dtype: int64
```

In [89]:
```
# Padronizando categorias relacionadas ao sexo

df_object.sexo.replace(to_replace= {'M':'Masculino','F':'Feminino','I':'Não Infor
df_object.sexo.value_counts()
```

```
c:\users\diego\appdata\local\programs\python\python37-32\lib\site-packages\pand
as\core\generic.py:6786: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stab
le/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydat
a.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y)
  self._update_inplace(new_data)
```

Out[89]:
```
Masculino        209855
Feminino          69478
Não Informado     52333
Name: sexo, dtype: int64
```

In [90]:
```
# Separando um DF para um estado

df_estado = df_object[df_object.uf == 'SP']
df_estado
```

Out[90]:

	uf	causa_principal	causa_acidente	tipo_acidente	classificacao_acidente	fase_dia	senti
0	SP	Sim	Falta de Atenção à Condução	Colisão com objeto estático	Com Vítimas Feridas	Plena Noite	Cre
1	SP	Sim	Falta de Atenção à Condução	Colisão com objeto estático	Com Vítimas Feridas	Plena Noite	Cre
2	SP	Sim	Falta de Atenção à Condução	Colisão com objeto estático	Com Vítimas Feridas	Plena Noite	Cre
3	SP	Sim	Falta de Atenção à Condução	Colisão com objeto estático	Com Vítimas Feridas	Plena Noite	Cre
34	SP	Não	Velocidade Incompatível	Colisão frontal	Com Vítimas Feridas	Plena Noite	Decre
...	...	...	...	...	...	...	
330990	SP	Sim	Não guardar distância de segurança	Colisão lateral	Com Vítimas Feridas	Pleno dia	Cre
330991	SP	Sim	Não guardar distância de segurança	Colisão lateral	Com Vítimas Feridas	Pleno dia	Cre
331175	SP	Sim	Falta de Atenção à Condução	Tombamento	Com Vítimas Feridas	Plena Noite	Cre
331176	SP	Sim	Falta de Atenção à Condução	Tombamento	Com Vítimas Feridas	Plena Noite	Cre
331177	SP	Sim	Falta de Atenção à Condução	Tombamento	Com Vítimas Feridas	Plena Noite	Cre

20574 rows × 15 columns

Selecionando Features

In [91]:
```
df_features = df_object.copy()
df_UF_features = df_estado.copy()

df_features.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 331666 entries, 0 to 331665
Data columns (total 15 columns):
uf                          331666 non-null object
causa_principal             331666 non-null object
causa_acidente              331666 non-null object
tipo_acidente               331666 non-null object
classificacao_acidente      331666 non-null object
fase_dia                    331666 non-null object
sentido_via                 331666 non-null object
condicao_metereologica      331666 non-null object
tipo_pista                  331666 non-null object
tracado_via                 331666 non-null object
uso_solo                    331666 non-null object
tipo_veiculo                331666 non-null object
tipo_envolvido              331666 non-null object
estado_fisico               331666 non-null object
sexo                        331666 non-null object
dtypes: object(15)
memory usage: 19.0+ MB
```

Transformando Categorias em Números

In [92]:
```
# Dados de todos os Estados

colunas = df_features.columns
colunas
```

Out[92]:
```
Index(['uf', 'causa_principal', 'causa_acidente', 'tipo_acidente',
       'classificacao_acidente', 'fase_dia', 'sentido_via',
       'condicao_metereologica', 'tipo_pista', 'tracado_via', 'uso_solo',
       'tipo_veiculo', 'tipo_envolvido', 'estado_fisico', 'sexo'],
      dtype='object')
```

In [93]:
```
# Todos os Estados

# Separa os dados que serão transformado num array
categorias = df_features.values

# Instancia o LabelEncoder()
labelEncoder_ = LabelEncoder()

# Transformando strings categóricas
for i in range(len(categorias[0])):
  categorias[:,i] = labelEncoder_.fit_transform(categorias[:,i])

df_categorico = pd.DataFrame(categorias, columns=colunas)
df_categorico
```

Out[93]:

	uf	causa_principal	causa_acidente	tipo_acidente	classificacao_acidente	fase_dia	sentid
0	25	1	11	4	1	2	
1	25	1	11	4	1	2	
2	25	1	11	4	1	2	
3	25	1	11	4	1	2	
4	17	1	11	8	1	2	
...	...	...	...	...	...	...	
331661	5	1	17	8	1	2	
331662	5	1	17	8	1	2	
331663	17	1	5	2	1	2	
331664	17	1	5	15	1	2	
331665	4	1	6	14	1	2	

331666 rows × 15 columns

In [94]:
```
# Dados de um estado

# Separa os dados que serão transformado num array
categorias_uf = df_UF_features.values

# Transformando stings categoricas
for i in range(len(categorias_uf[0])):
  categorias_uf[:,i] = labelEncoder_.fit_transform(categorias_uf[:,i])

df_categorico_uf = pd.DataFrame(categorias_uf, columns=colunas)
df_categorico_uf
```

Out[94]:

	uf	causa_principal	causa_acidente	tipo_acidente	classificacao_acidente	fase_dia	sentido_
0	0	1	11	4	1	2	
1	0	1	11	4	1	2	
2	0	1	11	4	1	2	
3	0	1	11	4	1	2	
4	0	0	23	5	1	2	
...	...	...	...	...	...	...	
20569	0	1	17	6	1	3	
20570	0	1	17	6	1	3	
20571	0	1	11	15	1	2	
20572	0	1	11	15	1	2	
20573	0	1	11	15	1	2	

20574 rows × 15 columns

Normalizando os Dados

In [95]:
```
# Todos os Estados

scaler = MinMaxScaler(feature_range = (0,1))
scaled_data = scaler.fit_transform(df_categorico)

df_categ_norm = pd.DataFrame(scaled_data, columns=colunas)
df_categ_norm
```

Out[95]:

	uf	causa_principal	causa_acidente	tipo_acidente	classificacao_acidente	fase_dia
0	0.961538	1.0	0.478261	0.2500	0.5	0.666667
1	0.961538	1.0	0.478261	0.2500	0.5	0.666667
2	0.961538	1.0	0.478261	0.2500	0.5	0.666667
3	0.961538	1.0	0.478261	0.2500	0.5	0.666667
4	0.653846	1.0	0.478261	0.5000	0.5	0.666667
...	...	...	...	...	...	...
331661	0.192308	1.0	0.739130	0.5000	0.5	0.666667
331662	0.192308	1.0	0.739130	0.5000	0.5	0.666667
331663	0.653846	1.0	0.217391	0.1250	0.5	0.666667
331664	0.653846	1.0	0.217391	0.9375	0.5	0.666667
331665	0.153846	1.0	0.260870	0.8750	0.5	0.666667

331666 rows × 15 columns

In [96]:
```
# Dados de um estado

scaled_data = scaler.fit_transform(df_categorico_uf)

df_categ_norm_uf = pd.DataFrame(scaled_data, columns=colunas)
df_categ_norm_uf
```

Out[96]:

	uf	causa_principal	causa_acidente	tipo_acidente	classificacao_acidente	fase_dia	sentid
0	0.0	1.0	0.478261	0.266667	0.5	0.666667	
1	0.0	1.0	0.478261	0.266667	0.5	0.666667	
2	0.0	1.0	0.478261	0.266667	0.5	0.666667	
3	0.0	1.0	0.478261	0.266667	0.5	0.666667	
4	0.0	0.0	1.000000	0.333333	0.5	0.666667	
...	...	...	...	...	...	...	
20569	0.0	1.0	0.739130	0.400000	0.5	1.000000	
20570	0.0	1.0	0.739130	0.400000	0.5	1.000000	
20571	0.0	1.0	0.478261	1.000000	0.5	0.666667	
20572	0.0	1.0	0.478261	1.000000	0.5	0.666667	
20573	0.0	1.0	0.478261	1.000000	0.5	0.666667	

20574 rows × 15 columns

Reduzindo Dimensionalidade

In [97]:
```
# Todos os Estados

pca=PCA(n_components=2)
pca.fit(df_categ_norm.values)
X_pca = pca.transform(df_categ_norm.values)
df_reduzido = pd.DataFrame(X_pca, columns=['x1', 'x2'])
df_reduzido

df_reduzido_dbscan = df_reduzido.copy()
```

In [98]:
```
# Dados de um estado

pca=PCA(n_components=2)
pca.fit(df_categ_norm_uf.values)
X_pca_uf = pca.transform(df_categ_norm_uf.values)
df_reduzido_uf = pd.DataFrame(X_pca_uf, columns=['x1-UF', 'x2-UF'])
df_reduzido_uf
```

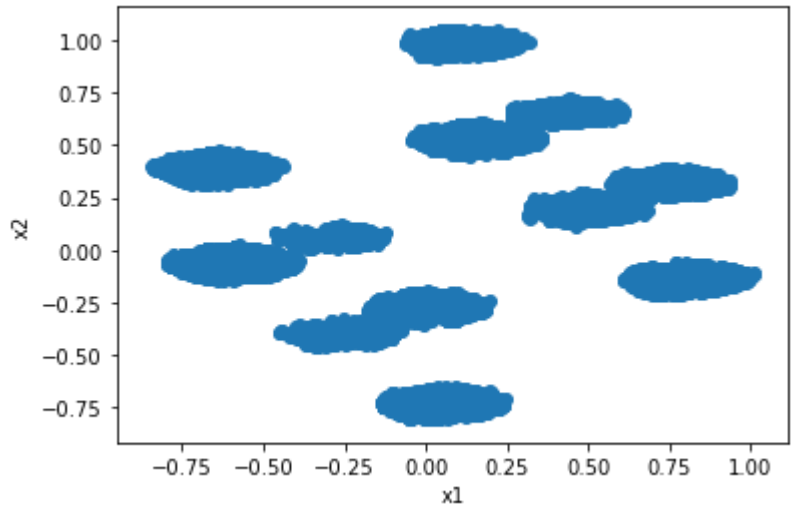
Out[98]:

	x1-UF	x2-UF
0	-0.431184	-0.236046
1	-0.431184	-0.236046
2	-0.444388	-0.240589
3	-0.457152	-0.251040
4	-0.663391	0.604541
...	...	...
20569	-0.342736	-0.134434
20570	-0.367908	-0.142556
20571	-0.275712	-0.118810
20572	-0.321635	-0.133622
20573	-0.360808	-0.153160

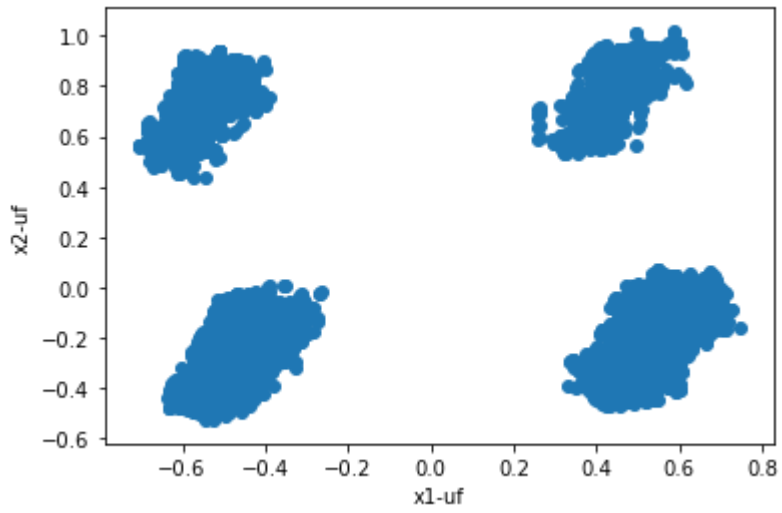
20574 rows × 2 columns

Visualizando os Dados

In [99]:
```
# Todos os Estados

plt.scatter(df_reduzido['x1'], df_reduzido['x2'])
plt.xlabel('x1')
plt.ylabel('x2')
plt.show()
```



In [100]:
```
# Dados de um estado = São Paulo

plt.scatter(df_reduzido_uf['x1-UF'], df_reduzido_uf['x2-UF'])
plt.xlabel('x1-uf')
plt.ylabel('x2-uf')
plt.show()
```



Executando Algorítmos

KMEANS

In [101]:
```
# Utilizando 12 Clusters, conforme visualizado no gráfico anterior

def getModel():
    return KMeans(n_clusters=12,random_state=0)
```

In [102]:
```
modelKmeans = getModel().fit(df_categ_norm.values)
modelKmeans.labels_
```

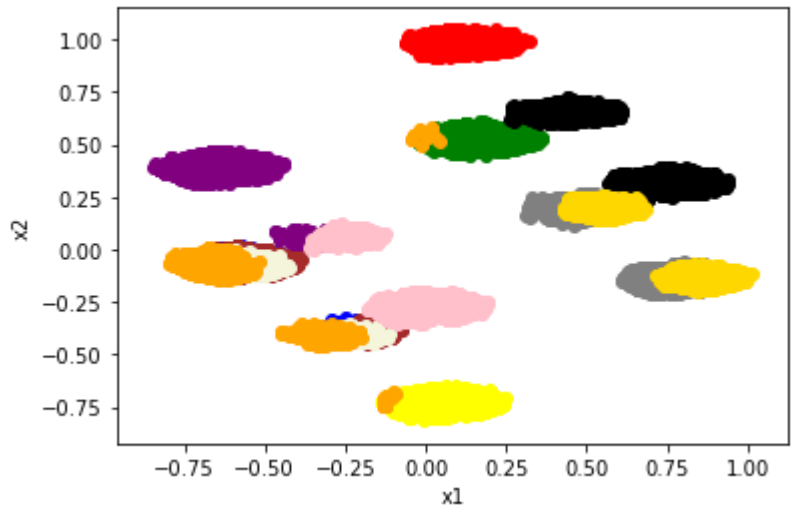
Out[102]:
```
array([2, 2, 9, ..., 0, 7, 7])
```

In [103]:
```
# Juntando o resultado ao DF

df_reduzido['rotulo'] = modelKmeans.labels_
df_reduzido
```

Out[103]:

	x1	x2	rotulo
0	0.458421	0.176235	2
1	0.458421	0.176235	2
2	0.479062	0.180016	9
3	0.484057	0.178230	9
4	0.834187	-0.124825	8
...	...	...	...
331661	0.818369	-0.125942	9
331662	0.842944	-0.119774	9
331663	-0.505150	-0.052482	0
331664	-0.514501	-0.073225	7
331665	-0.597596	-0.068525	7

In [104]:
```
# Visualizando resultado com cores

cores = ['blue','red','green','yellow','black', 'brown', 'purple', 'beige', 'gray

for i in range(len(df_reduzido['rotulo'].unique())):
    df_dados_cluster = df_reduzido[df_reduzido['rotulo'] == i]
    plt.scatter(df_dados_cluster['x1'],df_dados_cluster['x2'],color = cores[i])

plt.xlabel('x1')
plt.ylabel('x2')

plt.show()
```



Visualizando Resultados - KMEANS

In [105]:
```
df_object['rotulo'] = modelKmeans.labels_
df_object[df_object.rotulo == 2]
```

```
c:\users\diego\appdata\local\programs\python\python37-32\lib\site-packages\ipyk
ernel_launcher.py:1: 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/stab
le/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydat
a.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y)
  """Entry point for launching an IPython kernel.
```

Out[105]:

	uf	causa_principal	causa_acidente	tipo_acidente	classificacao_acidente	fase_dia	senti
0	SP	Sim	Falta de Atenção à Condução	Colisão com objeto estático	Com Vítimas Feridas	Plena Noite	Cre
1	SP	Sim	Falta de Atenção à Condução	Colisão com objeto estático	Com Vítimas Feridas	Plena Noite	Cre
9	SC	Sim	Ingestão de Substâncias Psicoativas	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
20	RS	Sim	Ingestão de Álcool	Colisão frontal	Com Vítimas Feridas	Plena Noite	Decre
22	RS	Sim	Ingestão de Álcool	Colisão frontal	Com Vítimas Feridas	Plena Noite	Decre
...	...	...	...	...	...	...	
331634	SC	Sim	Falta de Atenção à Condução	Colisão frontal	Com Vítimas Feridas	Pleno dia	Decre
331635	SC	Sim	Falta de Atenção à Condução	Colisão frontal	Com Vítimas Feridas	Pleno dia	Decre
331636	SC	Sim	Falta de Atenção à Condução	Colisão frontal	Com Vítimas Feridas	Pleno dia	Decre
331637	SC	Sim	Falta de Atenção à Condução	Colisão frontal	Com Vítimas Feridas	Pleno dia	Decre
331638	SC	Sim	Falta de Atenção à Condução	Colisão frontal	Com Vítimas Feridas	Pleno dia	Decre

39305 rows × 16 columns

In [106]:
```
df_object[df_object.rotulo == 8]
```

Out[106]:

	uf	causa_principal	causa_acidente	tipo_acidente	classificacao_acidente	fase_dia	ser
4	PR	Sim	Falta de Atenção à Condução	Colisão traseira	Com Vítimas Feridas	Plena Noite	C
14	GO	Sim	Desobediência às normas de trânsito pelo condutor	Colisão transversal	Com Vítimas Feridas	Plena Noite	Dec
108	RJ	Sim	Falta de Atenção à Condução	Tombamento	Com Vítimas Feridas	Pleno dia	C
109	RJ	Sim	Falta de Atenção à Condução	Colisão com objeto estático	Com Vítimas Feridas	Pleno dia	C
110	RJ	Sim	Falta de Atenção à Condução	Tombamento	Com Vítimas Feridas	Pleno dia	C
...	...	...	...	...	...	...	
331651	AC	Sim	Defeito na Via	Queda de ocupante de veículo	Com Vítimas Feridas	Amanhecer	C
331656	ES	Sim	Falta de Atenção à Condução	Tombamento	Com Vítimas Feridas	Pleno dia	C
331657	ES	Sim	Falta de Atenção à Condução	Tombamento	Com Vítimas Feridas	Pleno dia	C
331658	PR	Sim	Falta de Atenção à Condução	Colisão lateral	Com Vítimas Feridas	Pleno dia	Dec
331660	PR	Sim	Falta de Atenção à Condução	Colisão lateral	Com Vítimas Feridas	Pleno dia	Dec

32255 rows × 16 columns

DBScan

In [107]:
```
# Passando parametrização para algorítmo encontrar os cluster de forma independer

def getModelDbScan():
    return DBSCAN(eps=0.09 ,min_samples=30)
```

In [108]:
```
# Executando clusterrização com DBScan

modelDbscan = getModelDbScan().fit(df_categ_norm.values)
modelDbscan.labels_
```

Out[108]:
```
array([-1, -1, -1, ..., -1, -1, -1], dtype=int32)
```

In [109]:
```
# Verificando quantidade de clusters encontradas pelo algorítmo

a = set(modelDbscan.labels_)
len(a)
```

Out[109]:
```
102
```

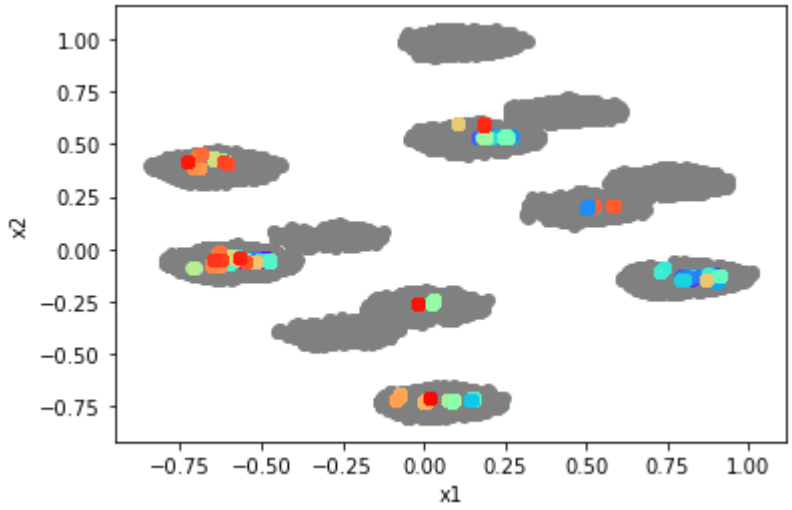
In [110]:
```
# Juntando resultado com o DF

df_reduzido_dbscan['rotulo'] = modelDbscan.labels_
df_reduzido_dbscan
```

Out[110]:

	x1	x2	rotulo
0	0.458421	0.176235	-1
1	0.458421	0.176235	-1
2	0.479062	0.180016	-1
3	0.484057	0.178230	-1
4	0.834187	-0.124825	-1
...	...	...	...
331661	0.818369	-0.125942	-1
331662	0.842944	-0.119774	-1
331663	-0.505150	-0.052482	-1
331664	-0.514501	-0.073225	-1
331665	-0.597596	-0.068525	-1

331666 rows × 3 columns

In [111]:
```
# Visualizando resultado com cores

colors = cm.rainbow(np.linspace(0, 1, len(df_reduzido_dbscan['rotulo'].unique()))

for i in df_reduzido_dbscan['rotulo'].unique():
    df_dados_cluster_dbscan = df_reduzido_dbscan[df_reduzido_dbscan['rotulo']==i

    if i== -1:
        plt.scatter(df_dados_cluster_dbscan['x1'],df_dados_cluster_dbscan['x2'],
    else:
        plt.scatter(df_dados_cluster_dbscan['x1'],df_dados_cluster_dbscan['x2'],


plt.xlabel('x1')
plt.ylabel('x2')

plt.show()
```



Visualizando Resultados - DBScan

In [112]:
```
df_object['rotulo'] = modelDbscan.labels_
df_object[df_object.rotulo == 2]
```

```
c:\users\diego\appdata\local\programs\python\python37-32\lib\site-packages\ipyk
ernel_launcher.py:1: 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/stab
le/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydat
a.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y)
  """Entry point for launching an IPython kernel.
```

Out[112]:

ificacao_acidente	fase_dia	sentido_via	condicao_metereologica	tipo_pista	tracado_via	uso_solo	ti
om Vítimas Feridas	Pleno dia	Crescente	Céu Claro	Simples	Reta	Sim	
om Vítimas Feridas	Pleno dia	Crescente	Céu Claro	Simples	Reta	Sim	C
om Vítimas Feridas	Pleno dia	Crescente	Céu Claro	Simples	Reta	Sim	
om Vítimas Feridas	Pleno dia	Crescente	Céu Claro	Simples	Reta	Sim	
om Vítimas Feridas	Pleno dia	Crescente	Céu Claro	Simples	Reta	Sim	C
...	...	...	...	...	...	...	
om Vítimas Feridas	Pleno dia	Crescente	Céu Claro	Simples	Reta	Sim	
om Vítimas Feridas	Pleno dia	Crescente	Céu Claro	Simples	Reta	Sim	
om Vítimas Feridas	Pleno dia	Crescente	Céu Claro	Simples	Reta	Sim	C
om Vítimas Feridas	Pleno dia	Crescente	Céu Claro	Simples	Reta	Sim	
om Vítimas Feridas	Pleno dia	Crescente	Céu Claro	Simples	Reta	Sim	

In [114]:
```
df_object[df_object.rotulo == 100]
```

Out[114]:

	uf	causa_principal	causa_acidente	tipo_acidente	classificacao_acidente	fase_dia	senti
331278	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331280	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331284	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331286	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331290	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331292	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331296	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331298	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331302	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331304	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331320	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331322	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331332	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331334	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331338	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331340	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331356	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331358	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331362	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331364	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331368	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331370	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre

	uf	causa_principal	causa_acidente	tipo_acidente	classificacao_acidente	fase_dia	senti
331374	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331376	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331380	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331382	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331398	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331400	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331410	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331412	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331476	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331478	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre
331488	MG	Sim	Velocidade Incompatível	Saída de leito carroçável	Com Vítimas Feridas	Plena Noite	Decre
331490	MG	Sim	Velocidade Incompatível	Tombamento	Com Vítimas Feridas	Plena Noite	Decre