

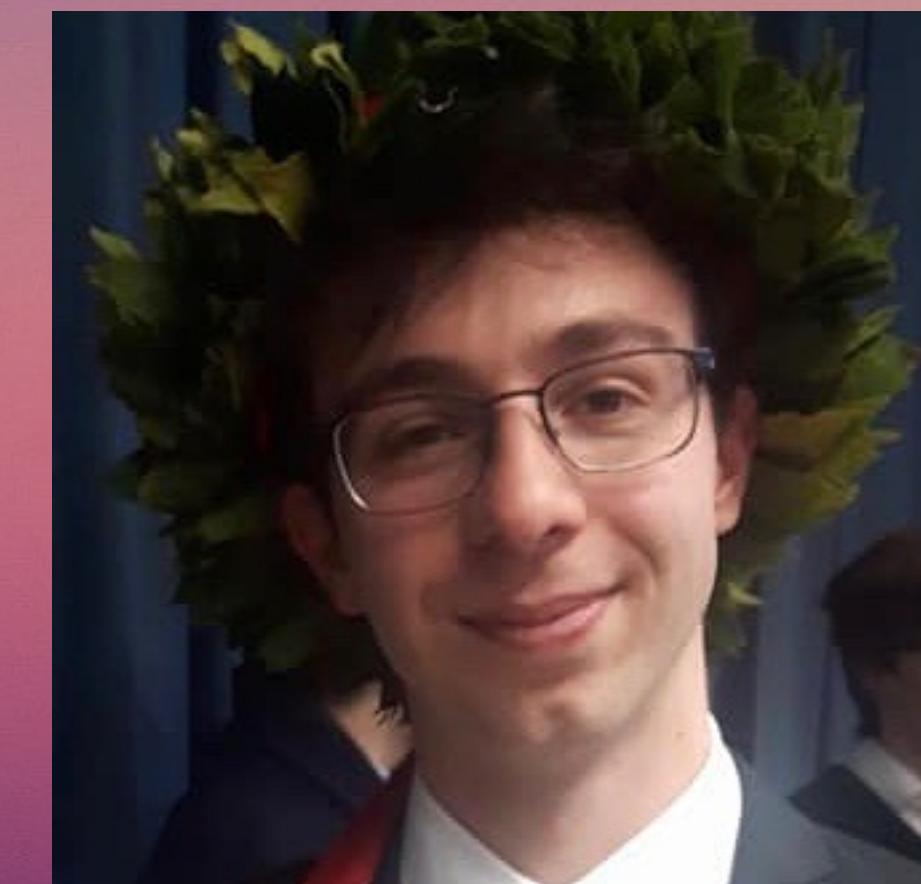


POLITECNICO  
MILANO 1863

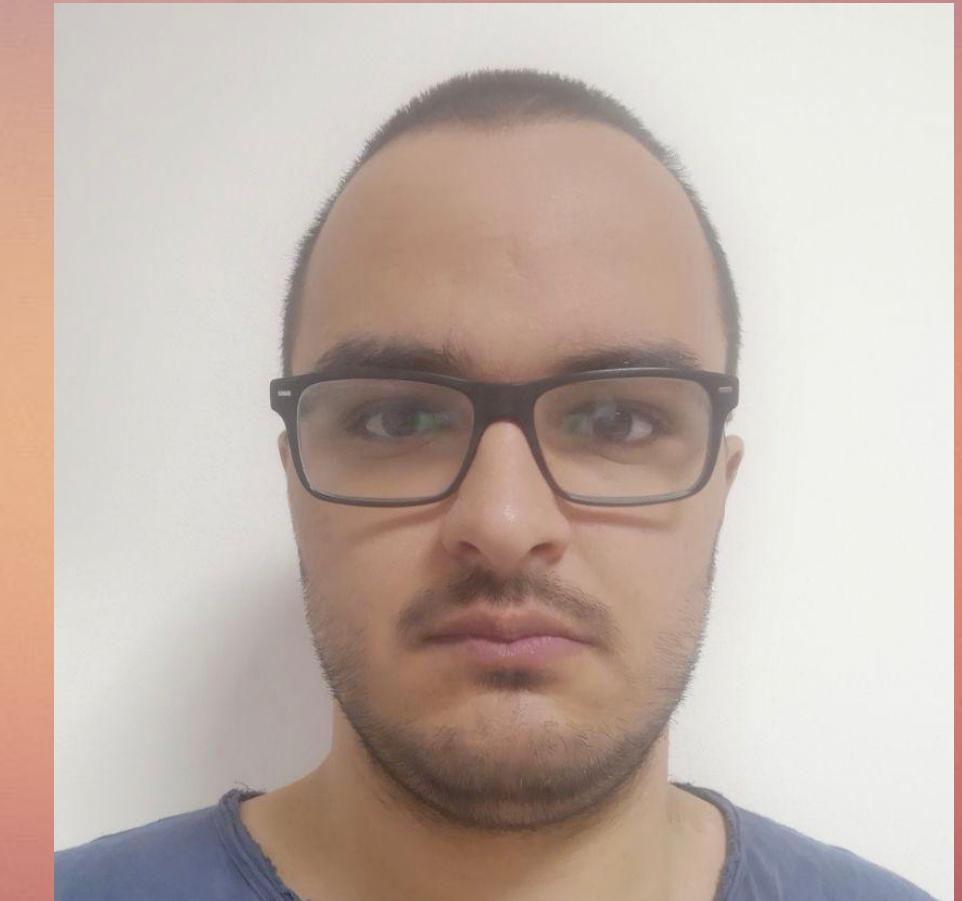
COMPUTER SCIENCE  
AND ENGINEER  
  
DATA AND  
INFORMATION QUALITY  
PROJECT

A.Y. 2018-2019

# DATA QUALITY ANALYZER



Leonardo Febbo



Giacomo Astolfi

# INTRODUCTION

PRESENTATION & DEVELOPMENT ENVIRONMENT

# DOMAIN PROBLEM

o o o

We have a large amount of data generated from a data stream of a transport company. In it there are informations regarding each validated ticket, on which bus and on which line.

# GOAL

...

The goal is to analyze the quality of this dataset and make some analysis on it and comparisons on portions of it.

We want to give users the opportunity to analyze these datasets based on their data quality dimensions and to make queries based on dataset attributes. The entire amount of data will be grouped in different datasets per day or week.

Additionaly, since the dataset is almost perfect, it will be "dirtied" in different ways on the different groups in order to analyze more cases of data quality problems.

# FEATURES

...

1. The user can choose one of the provided dataset he wants to analyze and build a query for it.

When provided with the query, the application will return the portion of the dataset which respects the requirements, and the user will be able to download it in CSV format. Some of the data will also be printed in order to provide feedback to the user.

2. In addition, the user will also receive as suggestions other portions of the dataset that perhaps only respect some of the requirements imposed by him and he will be able to decide whether to download even those or not.

# FEATURES

1. The user can choose one of the provided dataset he wants to analyze and build a query for it.

When provided with the query, the application will return the portion of the dataset which respects the requirements, and the user will be able to download it in CSV format. Some of the data will also be printed in order to provide feedback to the user.

The screenshot shows the 'Data Quality Analyzer' interface. At the top, there are tabs for 'Data Quality Analyzer', 'Query', and 'Compare'. Below this, there's a section for 'Select date' with buttons for 'Day 0' through 'Day 6'. To the right, under 'Select attributes to visualize', are buttons for CODLINHA, NOMELINHA, CODVEICULO, NUMEROCARTAO, DATAUTILIZACAO, COMPLETENESS, CONSISTENCY, and CONFORMITY. Below these are sections for 'Select only tuples with:' and 'Group by:', each containing dropdown menus for the same eight attributes. At the bottom, there's a 'Select groups with a group COUNT' dropdown and a 'On the attributes:' section with buttons for CODLINHA, NOMELINHA, CODVEICULO, NUMEROCARTAO, DATAUTILIZACAO, COMPLETENESS, CONSISTENCY, and CONFORMITY. Finally, at the very bottom are 'ADVANCED QUERY' and 'Query' buttons.

# FEATURES

o o o

2. In addition, the user will also receive as suggestions other portions of the dataset that perhaps only respect some of the requirements imposed by him and he will be able to decide whether to download even those or not.

The screenshot shows a user interface with a white background and three identical button templates stacked vertically. Each template includes a title, a descriptive subtitle, and a blue 'Compute Result' button. The first template is for 'COMPLETENESS', the second for 'CONSISTENCY', and the third for 'CONFORMITY'. Each template has a small gap between it and the next one.

Suggested queries:

Try the same query using only COMPLETENESS as quality dimension

Compute Result

Try the same query using only CONSISTENCY as quality dimension

Compute Result

Try the same query using only CONFORMITY as quality dimension

Compute Result

# FEATURES

...

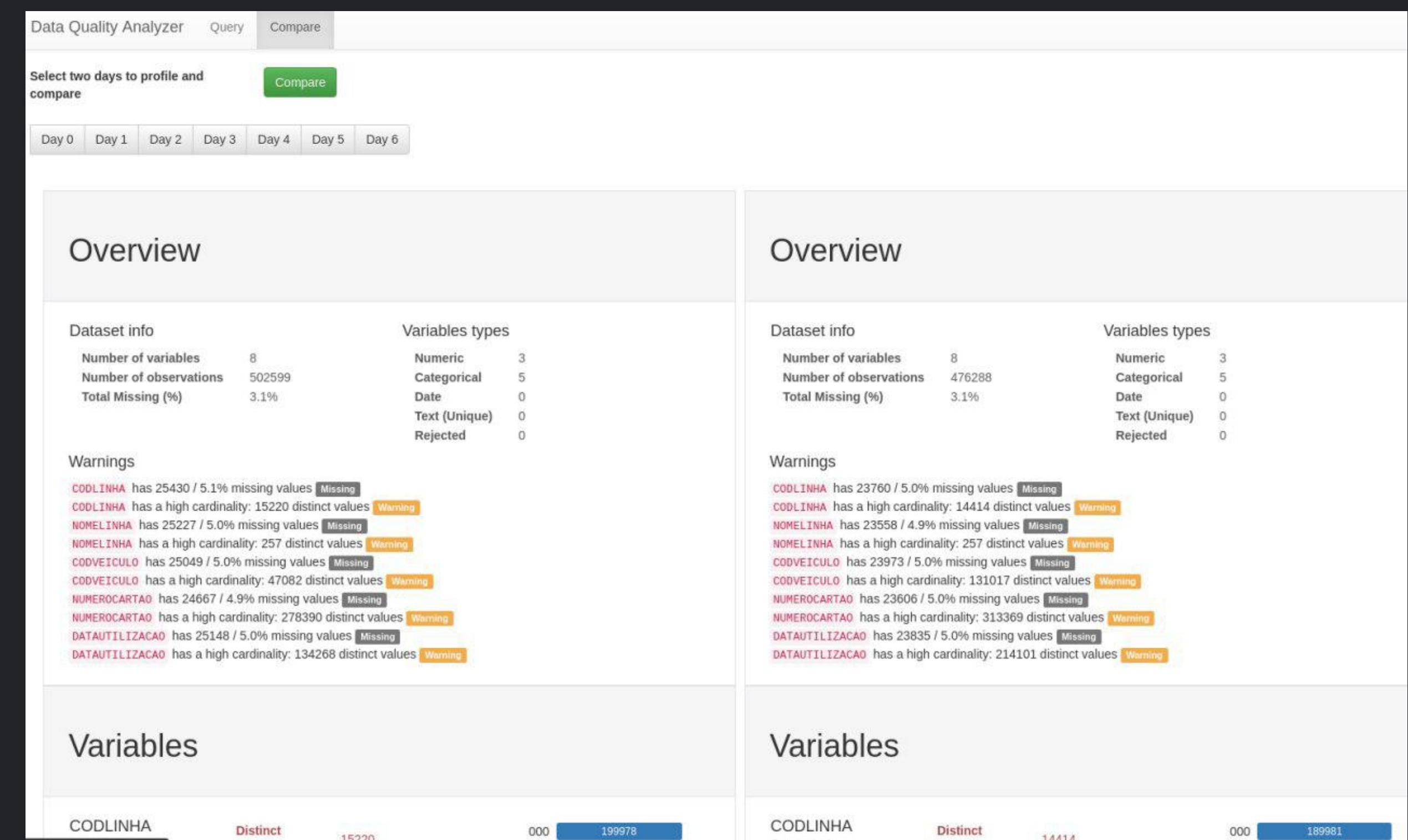
3. We also give the user the opportunity to see further details regarding each portion of the dataset obtained from the query (a sort of profiling) before downloading it.
  
4. The user can also compare two dataset using the profiling informations or compare two sections of them based on a query of his choice when possible.

# FEATURES

o o o

3. We also give the user the opportunity to see further details regarding each portion of the dataset obtained from the query (a sort of profiling) before downloading it.

4. The user can also compare two dataset using the profiling informations or compare two sections of them based on a query of his choice when possible.



# QUERY REQUIREMENTS

...

Completeness > value : **the user can search for data that has a completeness greater than the value entered**

Consistency > value : **the user can search for data that has a consistency greater than the value entered**

Timeliness > value : **the user can search for data that has a timeliness greater than the value entered**

Group by attributes : **the user can group the data based on a selected attributes**

Attributes: value : **the user can search for data that has a certain value for the attributes chosen**

# PROFILING REQUIREMENTS

...

Upon selecting a dataset, the application will show metadata about each column, including:

- The type (categorical, numerical, datetime etc.)
- The number and percentage of missing values
- The number and percentage of distinct values - A plot of the values (when possible)

And other information about the dataset itself, including:

- Simple metadata of the dataset (number of variables, size in memory, number of tuples, duplicate rows etc.)
- Mean and standard deviation of the completeness, timeliness and consistency
- Matrix plot of the missing values
- A random sample of the data

# COMPARE REQUIREMENTS

...

- The profiling information will be visible for both selected datasets
- The stats of the dataset with higher quality will be highlighted in some way (either by color or direct comparison)

•

# DEVELOPMENT ENVIRONMENT

o o o



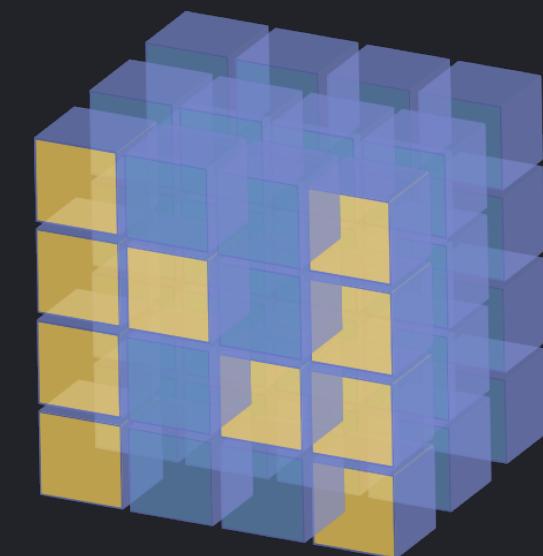
HTML



JS



CSS



NumPy

# DIRTY DATASET

HOW WE DUMPED THE DATASET

# DIRTY DATAFRAME

Parte Uno

...

Completeness

Parte Due

...

Consistency

Parte Tre

...

Conformity

# COMPLETENESS

○ ○ ○

```
df = df.mask(np.random.choice([True, False], size=df.shape, p=[.2, .8]))
```

	CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATAUTILIZACAO
0	280	N. SRA.DE NAZARÉ	BC911	1430250	07/10/15 07:37:02
1	280	N. SRA.DE NAZARÉ	BC911	NaN	07/10/15 07:51:25
2	NaN	N. SRA.DE NAZARÉ	BC911	NaN	NaN

# DIRTY DATAFRAME

Parte Uno

○ ○ ○

Completeness

Parte Due

○ ○ ○

Consistency

Parte Tre

○ ○ ○

Conformity

# CODLINHA

○○○

```
x = df[“CODLINHA”].to_numpy().copy()  
random.shuffle(x)  
  
import rstr  
for i in range(size_df):  
    new = rstr.xeger(r’(\d|[A-Z])\{3\}’)  
    x.extend([new])  
  
df[“CODLINHA”] = df[“CODLINHA”].mask(  
    np.random.choice([True, False], size=size_df, p=probability) == True, x)
```

	CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATAUTILIZACAO
0	OPC	N. SRA.DE NAZARÉ	BC911	1430250	07/10/15 07:37:02
1	000	N. SRA.DE NAZARÉ	BC911	2470195	07/10/15 07:51:25
2	280	N. SRA.DE NAZARÉ	BC911	3234514	07/10/15 18:49:49

# NOMELINHA

o o o

```
x = df[“NOMELINHA”].to_numpy().copy()  
random.shuffle(x)  
  
df[“NOMELINHA”] = df[“NOMELINHA”].mask(  
    np.random.choice([True, False], size=size_df, p=probability) == True, x)
```

	CODLINHA	NOMELINHA	CODVEICULO	NUMERO_CARTAO	DATA UTILIZACAO
0	280	OPER S/LINHA	BC911	1430250	07/10/15 07:37:02
1	280	OPER S/LINHA	BC911	2470195	07/10/15 07:51:25
2	280	VITÓRIA RÉGIA	BC911	3234514	07/10/15 18:49:49
3	280	V. IZABEL	BC911	3234514	07/10/15 18:49:51

# CODVEICULO

○ ○ ○

```
import rstr
x= []
for i in range(size_df):
    new = rstr.xeger(r'(\d\d|[A-Z][A-Z])(\d){3}')
    x.extend([new])

df["CODVEICULO"] = df[“CODVEICULO”].mask(
    np.random.choice([True, False], size=size_df, p=probability) == True, x)
```

	CODLINHA	NOMELINHA	CODVEICULO	NUMERO_CARTAO	DATA UTILIZACAO
0	280	N. SRA.DE NAZARÉ	BC911	1430250	07/10/15 07:37:02
1	280	N. SRA.DE NAZARÉ	EA332	2470195	07/10/15 07:51:25
2	280	N. SRA.DE NAZARÉ	BC911	3234514	07/10/15 18:49:49
3	280	N. SRA.DE NAZARÉ	TL272	3234514	07/10/15 18:49:51

# DATAUTILIZACAO

○ ○ ○

```
x = []
for i in range(size_df):
    new = rstr.xeger(r'(0[1-9]|12)[0-9]|3[01])/ (0[1-9]|1[0-2])/\d\d
          (00|0[0-9]|1[0-9]|2[0-3]):(0[0-9]| [0-5][0-9]):(0[0-9]| [0-5][0-9])')
    x.extend([new])

df["DATAUTILIZACAO"] = df["DATAUTILIZACAO"].mask(
    np.random.choice([True, False], size=size_df, p=probability) == True, x)
```

	CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATAUTILIZACAO
0	280	N. SRA.DE NAZARÉ	BC911	1430250	07/10/15 07:37:02
1	280	N. SRA.DE NAZARÉ	EA332	2470195	07/10/15 07:51:25
2	280	N. SRA.DE NAZARÉ	BC911	3234514	31/01/33 00:50:10
3	280	N. SRA.DE NAZARÉ	TL272	3234514	20/09/11 15:12:50

# DIRTY DATAFRAME

Parte Uno

○ ○ ○

Completeness

Parte Due

○ ○ ○

Consistency

Parte Tre

○ ○ ○

Conformity

# CODLINHA

○ ○ ○

```
x = []
for i in range(size_df):
    new = rstr.xeger(r'(\d|[A-Z])\{1,5\}')
    x.extend([new])

df["CODLINHA"] = df[“CODLINHA”].mask(
    np.random.choice([True, False], size=size_df, p=probability) == True, x)
```

	CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATAUTILIZACAO
3	280	N. SRA.DE NAZARÉ	BC911	0595898100	07/10/15 18:49:51
4	000	BAIRRO NOVO B	UV372	771305	07/05/16 00:04:09
5	DL4RG	OPER S/LINHA	08047	856665	07/10/15 16:50:18

# CODVEICULO

○ ○ ○

```
import rstr
x=[ ]
for i in range(size_df):
    new = rstr.xeger(r'(\d|[A-Z])\{0,1}(\d\d|[A-Z]|\d[A-Z])(\d)\{3\}(\d|[A-Z])\{0,1} ')
    x.extend([new])

df["CODVEICULO"] = df[“CODVEICULO”].mask(
    np.random.choice([True, False], size=size_df, p=probability) == True, x)
```

	CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATAUTILIZACAO
0	280	ABAETÉ	29073	1430250	07/10/15 07:37:02
1	280	N. SRA.DE NAZARÉ	HD128K	2470195	07/10/15 07:51:25
2	23A	OPER S/LINHA	BC911	3234514	07/10/15 18:49:49

# NUMEROCARTAO

○ ○ ○

```
x = []
for i in range(size_df):
    new = rstr.xeger(r'(\d){1,13}(\w){0,1}')
    x.extend([new])

df["NUMEROCARTAO"] = df["NUMEROCARTAO"].mask(
    np.random.choice([True, False], size=size_df, p=probability) == True, x)
```

CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATAUTILIZACAO
12	814	MOSSUNGUÊ	LN404	3527951
13	542	MOSSUNGUÊ	LN404	7324I
18	653	SABARÁ	HA293	120
19	653	SABARÁ	ND550	2471107

# DATAUTILIZACAO

○ ○ ○

```
#2015-10-07 19:30:20
for i in range(size_df):
    new = rstr.xeger(r'[12][0-9][0-9]-([0?][1-9]|1[0-2])-([0?][1-9]|1[2][0-9]|3[01])
                  ([00]|0[0-9]|1[0-9]|2[0-3]):([0[0-9]|[0-5][0-9]):([0[0-9]|[0-5][0-9])')
    x.extend([new])

#07-10-2015 19:30:20
for i in range(size_df):
    new = rstr.xeger(r'([0?][1-9]|1[2][0-9]|3[01])-([0?][1-9]|1[0-2])-\d\d
                  ([00]|0[0-9]|1[0-9]|2[0-3]):([0[0-9]|[0-5][0-9]):([0[0-9]|[0-5][0-9])')
    x.extend([new])
```

	CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATAUTILIZACAO
0	280	ABAETÉ	29073	1430250	07/10/15 07:37:02
1	280	N. SRA.DE NAZARÉ	HD128K	2470195	2948-02-4 00:0:0
2	23A	OPER S/LINHA	BC911	3234514	07/10/15 18:49:49

# DATA QUALITY DIMENSIONS

CHECK DATA QUALITY DIMENSIONS

# DATA QUALITY DIMENSIONS

Parte Uno

○ ○ ○

Completeness

Parte Due

○ ○ ○

Consistency

Parte Tre

○ ○ ○

Conformity

# COMPLETENESS

○ ○ ○

```
df['null-values'] = df.isnull().sum(axis=1)

conditions = [
    df['null-values'] == 0,
    df['null-values'] == 1,
    df['null-values'] == 2,
    df['null-values'] == 3,
    df['null-values'] == 4]
choices = [100, 80, 60, 40, 20]

df['COMPLETENESS'] = np.select(conditions, choices, default=0)

df = df.drop(['null-values'], axis=1)
```

	CODLINHA	NOMELINHA	CODEICULO	NUMEROCARTAO	DATAUTILIZACAO	COMPLETENESS
0	280	NaN	NaN	1430250	NaN	40
1	280	N. SRA.DE NAZARÉ	BC911	2470195	07/10/15 07:51:25	100
2	280	NaN	NaN	NaN	NaN	20
3	280	N. SRA.DE NAZARÉ	BC911	NaN	07/10/15 18:49:51	80

# DATA QUALITY DIMENSIONS

Parte Uno

○ ○ ○

Completeness

Parte Due

○ ○ ○

Consistency

Parte Tre

○ ○ ○

Conformity

# LIST LINHA

○○○

```
for i, row in df.iterrows():
    if (pd.isnull(row["CODLINHA"]) or pd.isnull(row["NOMELINHA"])):
        df.at[i, 'CHECK-LIST-LINHA'] = 0
    else:
        try:
            row_list_linha = list_linha.loc[row["CODLINHA"]]
            if row_list_linha["NOMELINHA"] == row["NOMELINHA"]:
                df.at[i, 'CHECK-LIST-LINHA'] = 1
            else:
                df.at[i, 'CHECK-LIST-LINHA'] = 0
        except:
            df.at[i, 'CHECK-LIST-LINHA'] = 0
```

	CODLINHA	NOMELINHA	CODVEICULO	NUMERO_CARTAO	DATA UTILIZACAO	COMPLETENESS	CHECK-LIST-LINHA
0	280	ABAETÉ	29073	1430250	07/10/15 07:37:02	100	0.0
1	280	N. SRA.DE NAZARÉ	HD128K	2470195	2948-02-4 00:0:0	100	1.0
2	23A	OPER S/LINHA	BC911	3234514	07/10/15 18:49:49	100	0.0

# CODLINHA

○○○

```

for i, row in df.iterrows():
    if pd.isnull(row["CODLINHA"]):
        df.at[i, 'CHECK-CODLINHA'] = 0
    else:
        try:
            row_list_linha = list_linha.loc[row["CODLINHA"]]
            df.at[i, 'CHECK-CODLINHA'] = 1
        except:
            df.at[i, 'CHECK-CODLINHA'] = 0

```

	CODLINHA	NOMELINHA	CODVEICULO	NUMERO_CARTAO	DATA UTILIZACAO	COMPLETENESS	CHECK-LIST-LINHA	CHECK-CODLINHA
0	280	ABAETÉ	29073	1430250	07/10/15 07:37:02	100	0.0	1.0
1	280	N. SRA.DE NAZARÉ	HD128K	2470195	2948-02-4 00:0:0	100	1.0	1.0
2	23A	OPER S/LINHA	BC911	3234514	07/10/15 18:49:49	100	0.0	0.0

# NOMELINHA

○○○

```
for i, row in df.iterrows():
    if pd.isnull(row["NOMELINHA"]):
        df.at[i, 'CHECK-NOMELINHA'] = 0
    else:
        try:
            row_list_linha = list_linha.loc[row["NOMELINHA"]]
            df.at[i, 'CHECK-NOMELINHA'] = 1
        except:
            df.at[i, 'CHECK-NOMELINHA'] = 0
```

	CODLINHA	NOMELINHA	CHECK-LIST-LINHA	CHECK-CODLINHA	CHECK-NOMELINHA
0	280	ABAETÉ	0.0	1.0	1.0
1	280	N. SRA.DE NAZARÉ	1.0	1.0	1.0
2	23A	OPER S/LINHA	0.0	0.0	1.0

# CODVEICULO

○○○

```
for i, row in df.iterrows():
    if pd.isnull(row["CODVEICULO"]):
        df_dirt.at[i, 'CHECK-CODVEICULO'] = 0
    else:
        try:
            row_list_vehicle = list_vehicle.loc[row["CODVEICULO"]]
            df_dirt.at[i, 'CHECK-CODVEICULO'] = 1
        except:
            df_dirt.at[i, 'CHECK-CODVEICULO'] = 0
```

	CODLINHA	NOMELINHA	CODVEICULO	CHECK-CODVEICULO
0	280	ABAETÉ	29073	0.0
1	280	N. SRA.DE NAZARÉ	HD128K	0.0
2	23A	OPER S/LINHA	BC911	1.0

# DATA UTILIZACAO

o o o

CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATA UTILIZACAO	CHECK-DATA
0	280	ABAETÉ	29073	1430250	07/10/15 07:37:02
1	280	N. SRA.DE NAZARÉ	HD128K	2470195	2948-02-4 00:0:0
2	23A	OPER S/LINHA	BC911	3234514	07/10/15 18:49:49

# CONSISTENCY

○ ○ ○

```

for i, row in df.iterrows():
    df.at[i, 'TOT'] = row["CHECK-LIST-LINHA"] +
        row["CHECK-CODLINHA"] + row["CHECK-NOMELINHA"] +
        row["CHECK-CODEICULO"] + row["CHECK-DATA"]

conditions = [
    df['TOT'] == 5.0,
    df['TOT'] == 4.0,
    df['TOT'] == 3.0,
    df['TOT'] == 2.0,
    df['TOT'] == 1.0]
choices = [100, 80, 60, 40, 20]
df['CONSISTENCY'] = np.select(conditions, choices, default=0)

```

	CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATAUTILIZACAO	COMPLETENESS	CONSISTENCY
0	280	ABAETÉ	29073	1430250	07/10/15 07:37:02	100	60
1	280	N. SRA.DE NAZARÉ	HD128K	2470195	2948-02-4 00:0:0	100	60
2	23A	OPER S/LINHA	BC911	3234514	07/10/15 18:49:49	100	60
3	280	N. SRA.DE NAZARÉ	435304P	0595898100	07/10/15 18:49:51	100	80
4	000	BAIRRO NOVO B	70X707M	771305	2567-11-20 14:56:9	100	40

# DATA QUALITY DIMENSIONS

Parte Uno

○ ○ ○

Completeness

Parte Due

○ ○ ○

Consistency

Parte Tre

○ ○ ○

Conformity

D Q A

# CODLINHA

o o o

```
def is_codlinha(str_datetime):  
    return re.search("^\d|[A-Z]\{3\}$", str_datetime)
```

	CODLINHA	NOMELINHA	CODEVEICULO	NUMEROCARTAO	DATAUTILIZACAO	CONF-CODLINHA
4	000	BAIRRO NOVO B	70X707M	771305	2567-11-20 14:56:9	1.0
5	DL4RG	OPER S/LINHA	08047	NaN	2619-4-31 09:21:35	0.0
6	NaN	ALTO BOQUEIRÃO	30918	2115218	07/10/15 13:18:26	0.0

# CODVEICULO

o o o

```
def is_codveiculo(str_datetime):
    return re.search("^\d\d[A-Z][A-Z]\d\d\d", str_datetime)
```

	CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATA UTILIZACAO	CONF-CODEVEICULO
0	280	ABAETÉ	29073	1430250	07/10/15 07:37:02	1.0
1	280	N. SRA.DE NAZARÉ	HD128K	2470195	2948-02-4 00:0:0	0.0
2	23A	OPER S/LINHA	BC911	3234514	07/10/15 18:49:49	1.0

# NUMEROCARTAO

○ ○ ○

```
def is_numerocartao(str_datetime):  
    return re.search("^\d{1,10}$", str_datetime)
```

# DATA UTILIZACAO

o o o

#07/10/15 19:30:20

```
def is_strdate(str_datetime):
    return re.search("^(0?[1-9]|1[0-9]|2[0-3])/(0?[1-9]|1[0-2])/\\d\\d
                    (00|0?[1-9]|1[0-9]|2[0-3]):(0?[1-9]|1[0-5]):(0?[1-9]|1[0-5])$",
                    str_datetime)
```

	CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATA UTILIZACAO	CONF-DATA UTILIZACAO
4	000	BAIRRO NOVO B	70X707M	771305	2567-11-20 14:56:9	0.0
5	DL4RG	OPER S/LINHA	08047	NaN	2619-4-31 09:21:35	0.0
6	NaN	ALTO BOQUEIRÃO	30918	2115218	07/10/15 13:18:26	1.0

# CONFORMITY

○ ○ ○

```

for i, row in df.iterrows():
    df.at[i, 'TOT'] = row["CONF-CODLINHA"] + row["CONF-CODEICULO"] +
    row["CONF-NUMEROCARTAO"] + row["CONF-DATAUTILIZACAO"]
conditions = [
    df['TOT'] == 4.0,
    df['TOT'] == 3.0,
    df['TOT'] == 2.0,
    df['TOT'] == 1.0]
choices = [100, 75, 50, 25]
df['CONFORMITY'] = np.select(conditions, choices, default=0)

```

	CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATAUTILIZACAO	COMPLETENESS	CONSISTENCY	CONFORMITY
0	280	ABAETÉ	29073	1430250	07/10/15 07:37:02	100	60	100
1	280	N. SRA.DE NAZARÉ	HD128K	2470195	2948-02-4 00:0:0	100	60	50
2	23A	OPER S/LINHA	BC911	3234514	07/10/15 18:49:49	100	60	100
3	280	N. SRA.DE NAZARÉ	435304P	0595898100	07/10/15 18:49:51	100	80	75
4	000	BAIRRO NOVO B	70X707M	771305	2567-11-20 14:56:9	100	40	50

# PROFILING

...

# OVERVIEW

o o o

## Overview

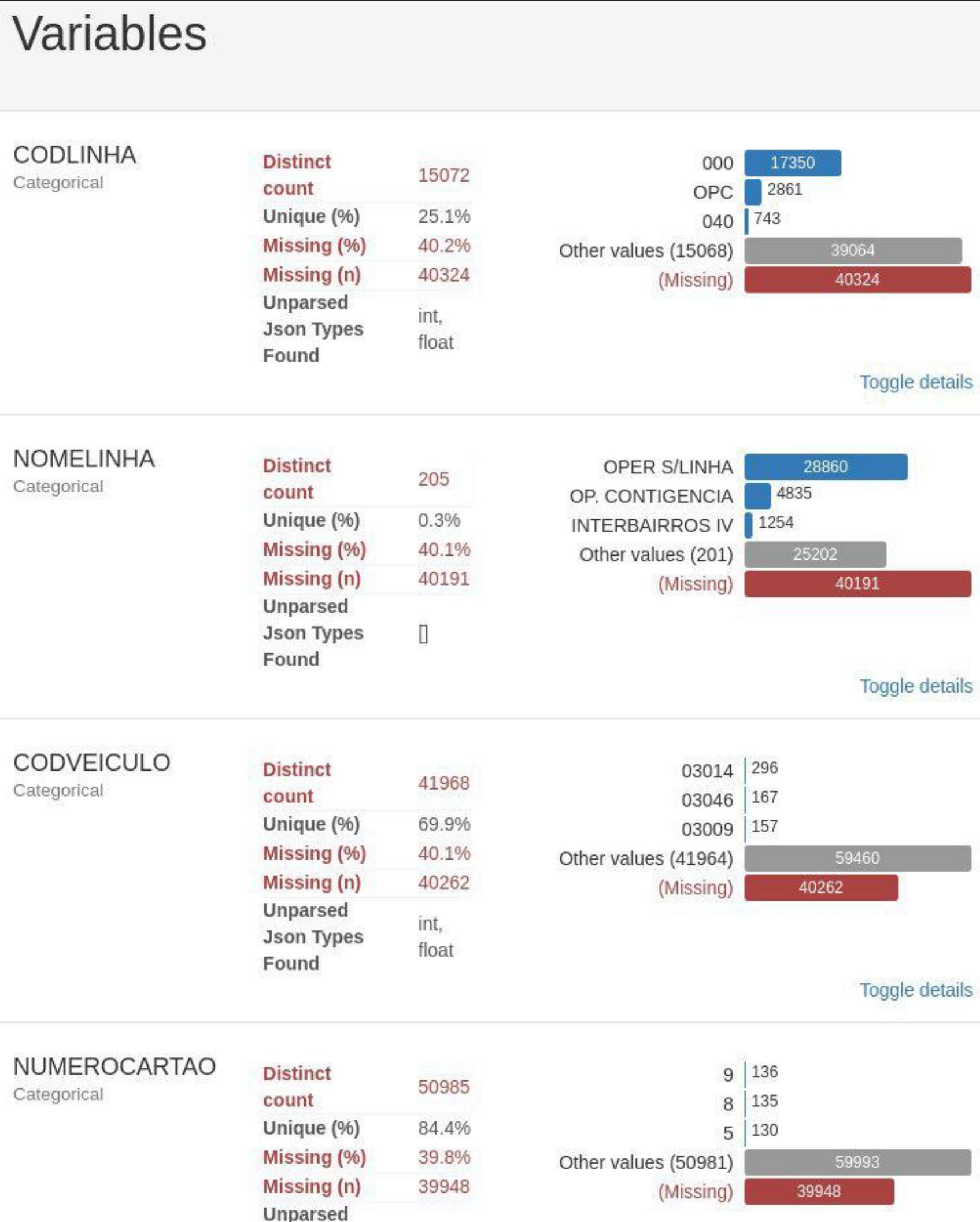
Dataset info		Variables types	
Number of variables	8	Numeric	3
Number of observations	100342	Categorical	5
Total Missing (%)	25.0%	Date	0
		Text (Unique)	0
		Rejected	0

### Warnings

- CODLINHA has 40324 / 40.2% missing values Missing
- CODLINHA has a high cardinality: 15072 distinct values Warning
- NOMELINHA has 40191 / 40.1% missing values Missing
- NOMELINHA has a high cardinality: 205 distinct values Warning
- CODVEICULO has 40262 / 40.1% missing values Missing
- CODVEICULO has a high cardinality: 41968 distinct values Warning
- NUMEROCARTAO has 39948 / 39.8% missing values Missing
- NUMEROCARTAO has a high cardinality: 50985 distinct values Warning
- DATAUTILIZACAO has 40055 / 39.9% missing values Missing
- DATAUTILIZACAO has a high cardinality: 53977 distinct values Warning
- COMPLETENESS has 1011 / 1.0% zeros
- CONSISTENCY has 12096 / 12.1% zeros
- CONFORMITY has 12978 / 12.9% zeros

# VARIABLES

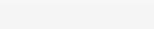
o o o



D Q A

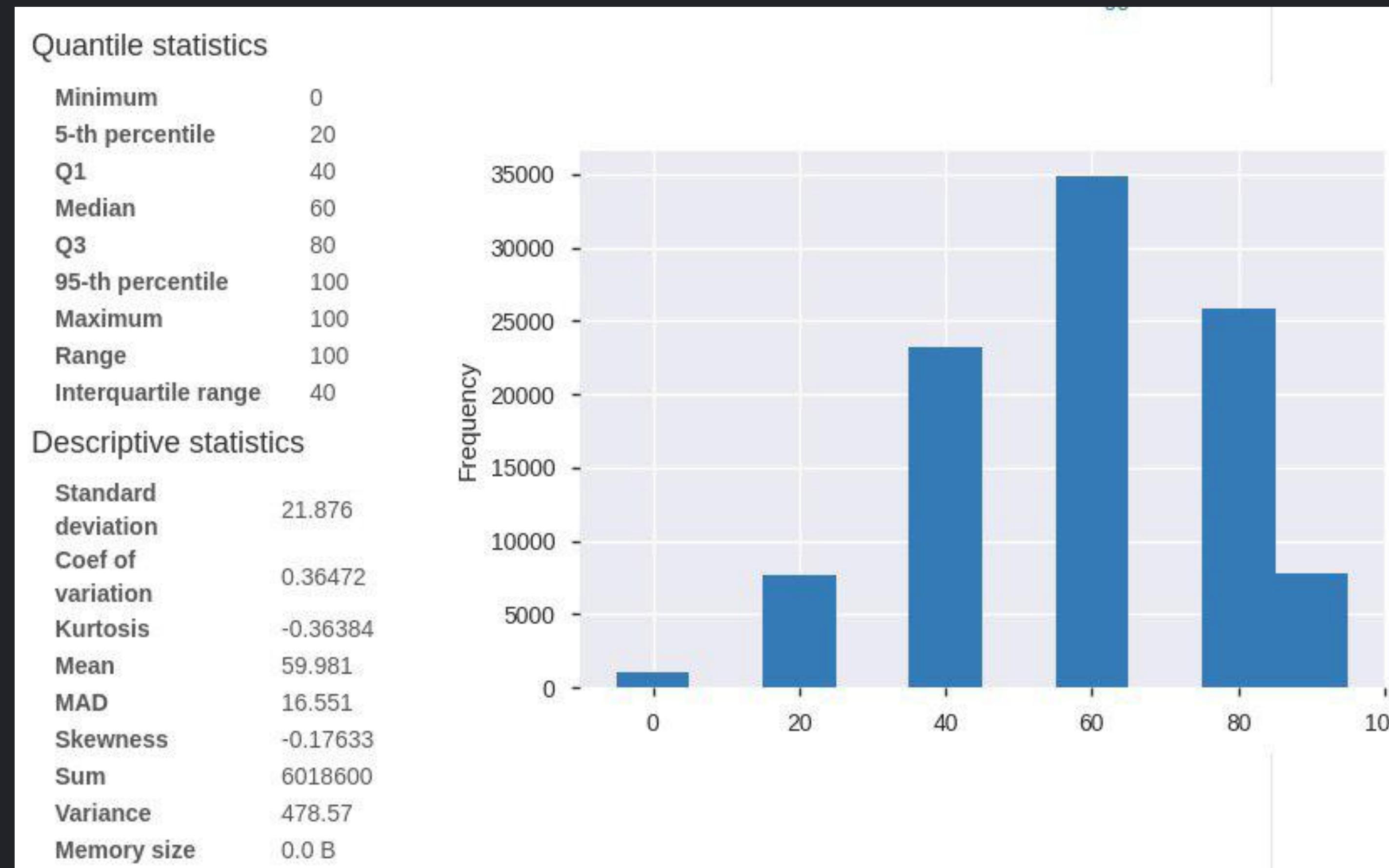
# DETAILS

o o o

NOMELINHA	Distinct count	205
	Unique (%)	0.3%
	Missing (%)	40.1%
	Missing (n)	40191
	Unparsed	
	Json Types Found	[]
Toggle details		
Value	Count	Frequency (%)
OPER S/LINHA	28860	28.8% 
OP. CONTIGENCIA	4835	4.8% 
INTERBAIRROS IV	1254	1.2% 
SISTEMA ARAUCARIA	828	0.8% 
INTERBAIRR II H	787	0.8% 
INTERBAIRROS III	683	0.7% 
INTERB II ANTI H	620	0.6% 
INTERBAIRROS V	558	0.6% 
RIO BONITO	514	0.5% 
S.RITA/PINHEIRIN	503	0.5% 
CABRAL / PORTÃO	479	0.5% 
C.RASO/CAIUÁ	439	0.4% 
CENTENÁRIO/HAUER	404	0.4% 
TERMINAL PINHEIRINHO	354	0.4% 
ALTO BOQUEIRÃO	317	0.3% 
C.COMP/C.RASO	297	0.3% 
BARREIRINHA	259	0.3% 

# DETAILS

o o o



# SAMPLES

o o o

## Sample

	CODLINHA	NOMELINHA	CODVEICULO	NUMEROCARTAO	DATAUTILIZACAO	COMPL
0	170	None	7D4642	None	2833-12-30 10:05:2	60
1	000	None	00186	None	31/02/23 00:00:02	60
2	P26I	INTERBAIRROS V	None	None	None	40
3	None	SISTEMA ARAUCARIA	None	None	None	20
4	000	None	None	None	25/10/15 16:31:19	40

# DEMO