

IMD0033 - Probabilidade

Aula 07 - Higienização, Imputação e Pivoteamento de Dados

Ivanovitch Silva
Março, 2019






Introdução a Pandas

Explorando dados com Pandas

Imputação, higienização e limpeza da dados

Revisão

Prova #01

- 
- Estudo de caso: Titanic
 - Imputação de dados
 - Higienização dos dados
 - Pivoteamento de tabelas
 - Limpando dados faltantes

Atualizar o repositório

```
git clone https://github.com/ivanovitchm/imd0033_2019_1
```

Ou

```
git pull
```



Estudo de Caso: Titanic



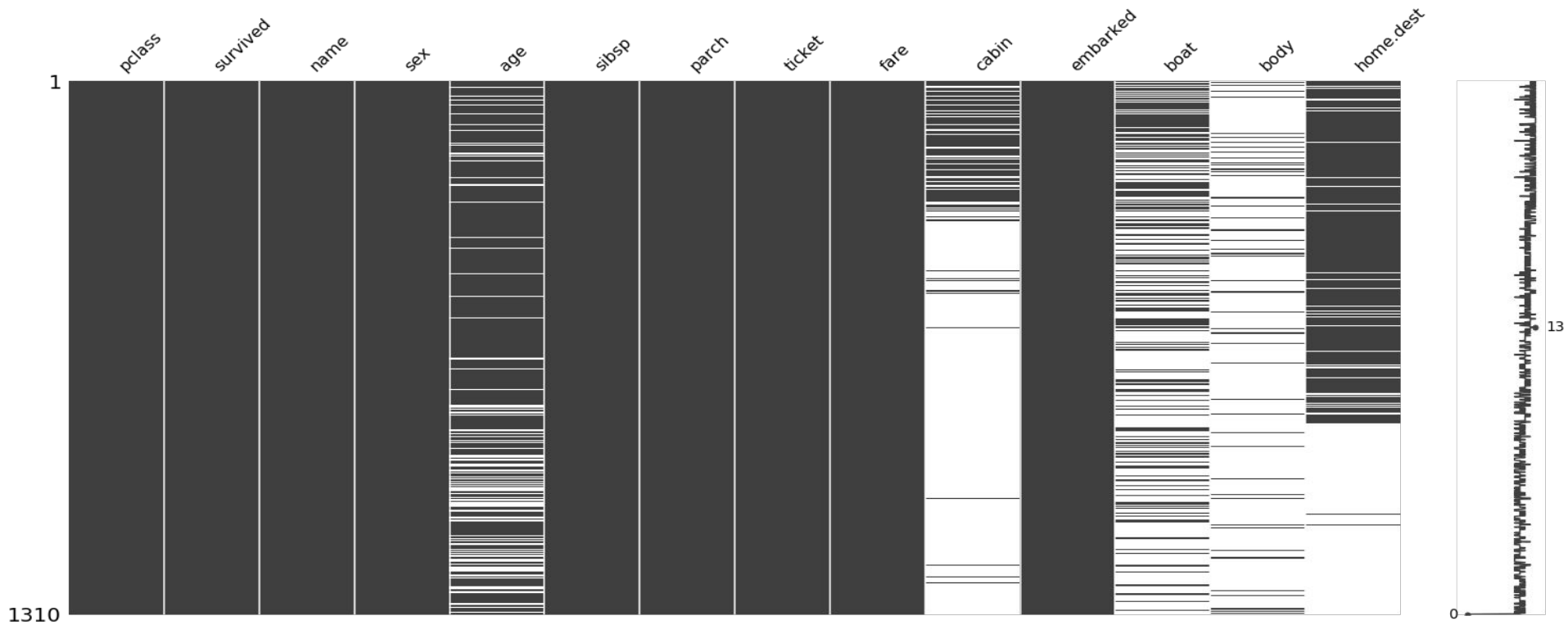
kaggle

<https://www.kaggle.com/c/titanic>

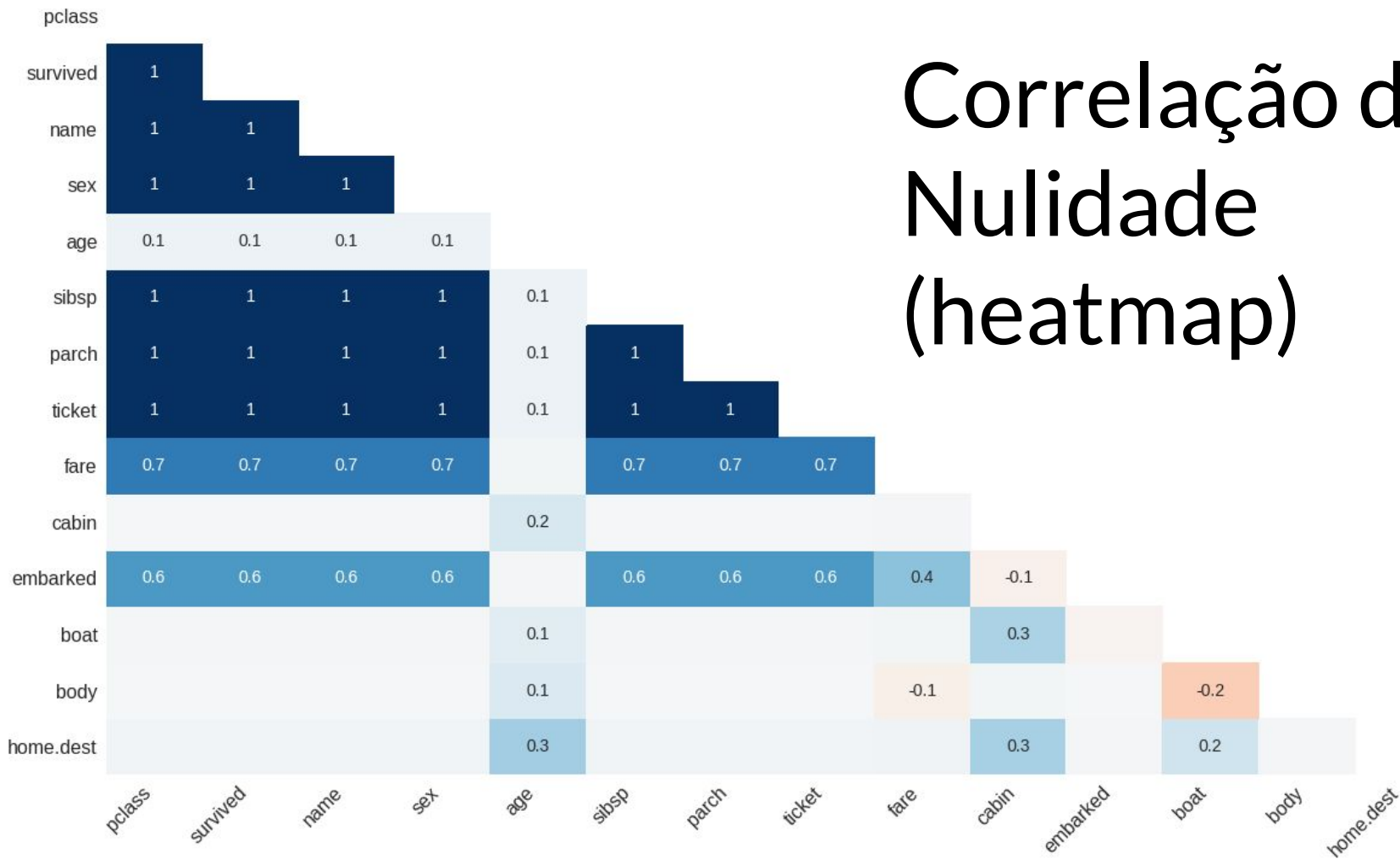
Estudo de caso: Titanic

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S	2		St Louis, MO
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S	11		Montreal, PQ / Chesterville, ON
2	1	0	Allison, Miss. Helen Loraine	female		2	1	2	113781	151.5500	C22 C26	S		Montreal, PQ / Chesterville, ON
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	C22 C26	S		135	Montreal, PQ / Chesterville, ON
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25	1	2	113781	151.5500	C22 C26	S			Montreal, PQ / Chesterville,)

Visualizando dados faltantes (matrix)



Correlação de Nulidade (heatmap)



Imputação de dados

O que esses códigos fazem?

```
titanic_survival.loc[~titanic_survival.age.isnull(), "age"].shape  
titanic_survival[~titanic_survival["age"].isnull() ].shape
```

Qual o ponto de discussão sobre imputação?

```
mean_age = sum(titanic_survival["age"]) / len(titanic_survival["age"])
```

Qual o valor da variável "mean_age" se algum valor da coluna "age" estiver faltando?

Algumas facilidades da API Pandas

```
correct_mean_age = titanic_survival["age"].mean()
```

Com sorte, a imputação de dados é bastante comum e uma grande maioria de métodos na API Pandas já realiza o filtro de dados faltantes.

Desafio

Qual o valor médio das passagens por classe?

- Exercício seção 5

Qual a idade média dos passageiros por classe?

Lesson 07 - Advanced Pandas.ipynb Up to Section 5



Calculando estadísticas descriptivas

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1.0	1.0	Allen, Miss. Elisabeth Walton	female	29.0000	0.0	0.0	24160	211.3375	B5	S	2	NaN	St Louis, MO
1	1.0	1.0	Allison, Master. Hudson Trevor	male	0.9167	1.0	2.0	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON
2	1.0	0.0	Allison, Miss. Helen Loraine	female	2.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON

```
fares_by_class = {i:titanic_survival[titanic_survival.pclass == i].fare.mean()
                  for i in titanic_survival.pclass.unique()
                  }
```

{1.0: 87.50899164086687, 2.0: 21.1791963898917, 3.0: 13.302888700564957}

Pivoteamento de tabelas

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S	2		St Louis, MO
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S	11		Montreal, PQ / Chesterville, ON
2	1	0	Allison, Miss. Helen Loraine	female	2	1	2	113781	151.5500	C22 C26	S			Montreal, PQ / Chesterville, ON

```
passenger_class_fares = titanic_survival.pivot_table(index="pclass",
values="fare", aggfunc=np.mean)
```

```
import numpy as np
df_pivot = titanic_survival.pivot_table(index="pclass",
                                         values=["fare", "age"],
                                         aggfunc=[np.mean, len])
```

	mean		len	
	age	fare	age	fare
pclass				
1.0	39.159918	87.508992	323.0	323.0
2.0	29.506705	21.179196	277.0	277.0
3.0	24.816367	13.302889	709.0	709.0

```
df_pivot["mean"]["age"][1.0]
39.159918
```

Desafio

Qual a percentagem de sobreviventes para grupos de diferentes idades?

- 0 - 5 (infantil)
- 6 - 10 (criança)
- 11 - 18 (adolescente)
- 19 - 30 (adulto jovem)
- 31 - 50 (adulto pleno)
- 51 - 65 (adulto senior)
- 66 - (idoso)

		count
agecat	survived	
Infant	0.0	19
	1.0	37
Child	0.0	17
	1.0	13
Teenager	0.0	62
	1.0	45
Young adult	0.0	263
	1.0	153
Adult	0.0	201
	1.0	141
Senior adult	0.0	49
	1.0	36
Senior	0.0	8
	1.0	2

Outra forma de agregação

```
titanic_survival["agecat"] = pd.cut(titanic_survival.age,  
                                     bins=[0,5,10,18,30,50,65,100],  
                                     labels=["Infant","Child","Teenager",  
                                             "Young adult","Adult","Senior adult","Senior"])
```

	agecat	age
0	Young adult	29.0000
1	Infant	0.9167
2	Infant	2.0000
3	Young adult	30.0000
4	Young adult	25.0000
5	Adult	48.0000


```
titanic_survival.pivot_table(index=["agecat", "survived"],  
                              values="age",  
                              aggfunc="count").
```



```
aggfunc=lambda x: len(x)/len(titanic_survival[~titanic_survival.age.isnull()])
```



```
index.js
import React, { useState } from 'react';
import './index.css';
import './index.html';
import './index.js';

function App() {
  const [contacts, setContacts] = useState([]);
  const [name, setName] = useState('');
  const [phone, setPhone] = useState('');
  const [email, setEmail] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    setContacts([...contacts, { name, phone, email }]);
    setName('');
    setPhone('');
    setEmail('');
  };

  return (
    <div>
      <h1>React Form</h1>
      <form>
        <input type="text" value={name} onChange={e => setName(e.target.value)} />
        <input type="text" value={phone} onChange={e => setPhone(e.target.value)} />
        <input type="text" value={email} onChange={e => setEmail(e.target.value)} />
        <button type="button" value="Submit" />
      </form>
      <ul>
        {contacts.map((contact) => (
          <li>
            {contact.name} {contact.phone} {contact.email}
          </li>
        ))}
      </ul>
    </div>
  );
}

export default App;
```

```
index.html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>React Form</title>
  </head>
  <body>
    <div>
      <h1>React Form</h1>
      <form>
        <input type="text" value="" />
        <input type="text" value="" />
        <input type="text" value="" />
        <button type="button" value="Submit" />
      </form>
      <ul>
        <li></li>
      </ul>
    </div>
  </body>
</html>
```