


# IMD0033 - Probabilidade

## Aula 11 - Pandas Avançado

Ivanovitch Silva  
Setembro, 2018



# Agenda

---

- 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_2018_2.git
```

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, )

# 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 11 - Advanced Pandas Working with Missing Data.ipynb

## Up to section 5



# 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)
```

# Limpendo os dados faltantes

```
print(df.dropna())
```

	A	B	C	D
w	6.0	3.0	7.0	4.0
y	4.0	3.0	7.0	7.0

```
print(df.dropna(axis=1))
```

	A	B	D
w	6.0	3.0	4.0
x	6.0	2.0	7.0
y	4.0	3.0	7.0
z	2.0	5.0	1.0

	A	B	C	D
w	6.0	3.0	7.0	4.0
x	6.0	2.0	NaN	7.0
y	4.0	3.0	7.0	7.0
z	2.0	5.0	NaN	1.0

# 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)

		age
agecat	survived	
Infant	0.0	0.018164
	1.0	0.035373
Child	0.0	0.016252
	1.0	0.012428
Teenager	0.0	0.059273
	1.0	0.043021
Young adult	0.0	0.251434
	1.0	0.146272
Adult	0.0	0.192161
	1.0	0.134799
Senior adult	0.0	0.046845
	1.0	0.034417
Senior	0.0	0.007648
	1.0	0.001912

```
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"])
```

```
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';

function App() {
  const [contacts, setContacts] = useState([
    { name: 'John Doe', phone: '123-456-7890' },
    { name: 'Jane Smith', phone: '987-654-3210' }
  ]);

  const handleClick = () => {
    // TODO: Implement handleClick logic
  };

  return (
    <div>
      <h1>Contact List</h1>
      <ul>
        {contacts.map(contact => (
          <li>{contact.name} {contact.phone}</li>
        ))}
      </ul>
      <button onClick={handleClick}>Add Contact</button>
    </div>
  );
}

export default App;
```

```
index.html
<!DOCTYPE html>
<html>
  <head>
    <script src="index.js"></script>
  </head>
  <body>
    <div></div>
  </body>
</html>
```