

# Exploratory Data Analysis of Titanic data set

Fyodor Raevskiy<sup>1</sup>

<sup>1</sup>January 2022

\*For correspondence:

[xboxraevskii@mail.ru](mailto:xboxraevskii@mail.ru) (FMS);

[@iwarnedyouaboutstairss](https://www.instagram.com/iwarnedyouaboutstairss)

(Telegram)

<sup>†</sup>Project for Tinkoff Generation

<sup>‡</sup>The data set was taken from [kaggle.com](https://www.kaggle.com)

**Abstract** In this research I want to analyse information about people on Titanic , we will understand who has survived and who has deceased , make some hypothesises on this topic and try to prove it by statistics.

## Introduction

Hello! This is my first EDA so all calculations and conclusions I will do step by step and show you how I get them. We will be working with the Titanic Data Set from Kaggle. This is a very famous data set and very often is a student's first step in Exploratory Data Analysis and machine learning!

Firstly , we need to import some libraries that will help us in analysing data.

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 %matplotlib inline
```

## Reading the data

One of the most important first steps is to understand what data do we have. Let's start by reading in the "titanic train.csv" file into a pandas data frame. There are a lot of functions that generally return a pandas object, but in our case, we will use `pandas.read_csv()` which is the most popular at newbies.

```
1 train = pd.read_csv("C:/datasets/train.csv")
```

## How our data looks like

The easiest way to look on your data set is to show just a few of first lines of our data set. To do it let's print first 5 strings.

```
1 train.head()
```

What does each column mean?

Okay, now we have seen our information but I guess that reader needs some explanations about some of it.

Let's start with P-class. That column shows us in which class passenger was.

Then we can see this weird column which called parch. It shows number of brothers, sisters, step-brothers, stepsisters, spouses on board the Titanic.

Everyone understands what do fare and cabin mean but I guess not even a majority understood what does Embarked mean. 'Embarked' shows port of embarkation. Unexpectedly? 'C' is for Cherbourg , 'S' is for Southampton and 'Q' is for Queenstown.

**Table 1.** First 12 lines of Titanic data set.

Pass-ID	Surv	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
9	1	3	Johnson, Mrs. Oscar W	female	27.0	0	2	347742	11.1333	NaN	S
10	1	2	Nasser, Mrs. Nicholas	female	14.0	1	0	237736	30.0708	NaN	C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S

Source: <https://www.kaggle.com/hesh97/titanicdataset-traincsv>

### 39 Missing data

40 As we can see on our table , there are a lot of missing points about the cabin and the age of a  
 41 passenger. And I would choose to rid of it but let me explain why. Getting rid of NaN objects in  
 42 most cases caused simplicity. As data comes in many shapes and forms, we aim to find the easiest  
 43 way of understanding statistics. We can use seaborn to create a simple heat map to see where we  
 44 are losing information.

```
45 1 train.isnull()
```

46 This function is very easy , it just checks our dataframe and shows True if this parameter is NaN  
 47 and shows False if it's not.

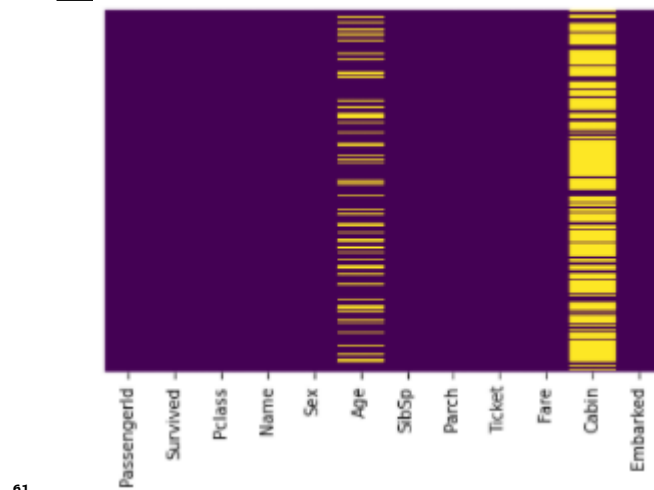
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
	0	False	False	False	False	False	False	False	False	False	True	False
	1	False	False	False	False	False	False	False	False	False	False	False
	2	False	False	False	False	False	False	False	False	False	True	False
	3	False	False	False	False	False	False	False	False	False	False	False
	4	False	False	False	False	False	False	False	False	False	True	False
	...	...	...	...	...	...	...	...	...	...	...	...
	886	False	False	False	False	False	False	False	False	False	True	False
	887	False	False	False	False	False	False	False	False	False	False	False
	888	False	False	False	False	True	False	False	False	False	True	False
	889	False	False	False	False	False	False	False	False	False	False	False
	890	False	False	False	False	False	False	False	False	False	True	False

48  
 49 For example , the first cell of Cabin is True , that means that we have no information about first  
 50 passenger's cabin.

51 But as you already understood this is not the best way to remove missing data , cause it become  
 52 more difficult if we have tons of information. So I offer you to use another method , method of  
 53 visualisation . Let 's create a graphic that will show us which column has the most of missing data.

```
54 1 sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
55 2 #Some explanations: when I typed train.isnull() in brackets I said to sns library that it
56     should take train.isnull() and whenever it's true sns will display it in another color
57     on graphic. The y axis is just all passengers but I typed yticklabels=False cause I
58     do not want to see all of them.And the last two arguments are just for graph's
59     appearance.
```

60 *isNull graphic:*



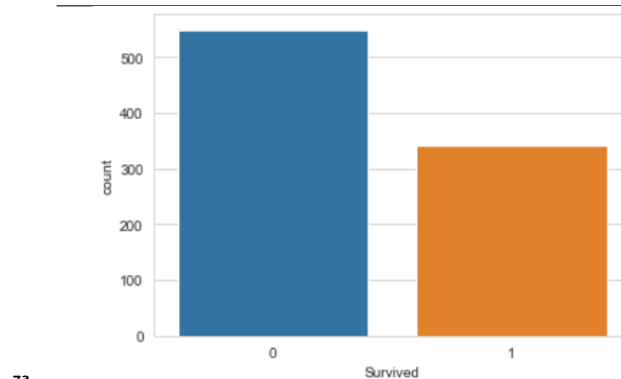
62 **What we are supposed to do?**

63 So, almost 20 percent of the data on the age of passengers is missing. But it seems to me that such  
64 a proportion is reasonable enough to replace this data with something sane. But we seem to have  
65 no information about the cabins at all , and therefore we will most likely get rid of this column or  
66 replace it with "Is there information about the cabin: 1 for yes and 0 for no"

67 **Let's continue visualising some more of the data.**

68 Let's find out how many people survived on Titanic

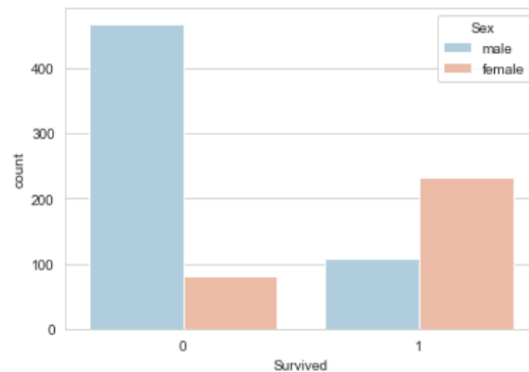
```
69 1 sns.set_style('whitegrid') #it will create beautiful grid
70 2 sns.countplot(x='Survived',data=train)
71 3 #Some explanations : depending on Survived column I will create a graph that will show how
72 many people died (how many 0 does 'survived' column have) and how many people survived
```



73

74 As you can see a lot of people did not survived , I would rather say the majority of passengers  
75 did not survived.Let's see did more men or women survive?

```
76 1 sns.countplot(x='Survived',hue='Sex',data=train,palette='RdBu_r')
```



77

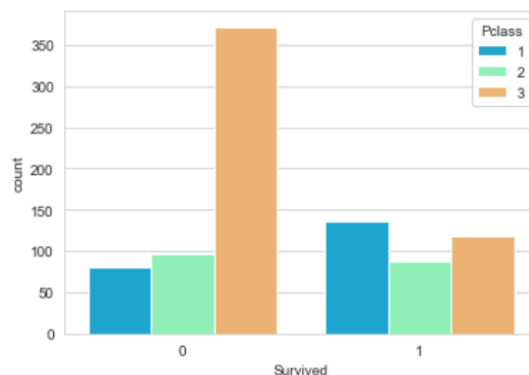
78 So most of men died and at least 80 women also died.Yes , it's a pity , but there's nothing we can  
79 do about it.

## 80 Comparing different classes

81 Let's see which class of passengers has survived the most and which the least.

```
82 1 sns.countplot(x='Survived',hue='Pclass',data=train,palette='rainbow')
```

```
83 2 #Some explanations: now I typed to sbs that it must take column 'Survived' and count,  
84     depending on column 'Pclass', how many passengers of each class have survived.
```



85

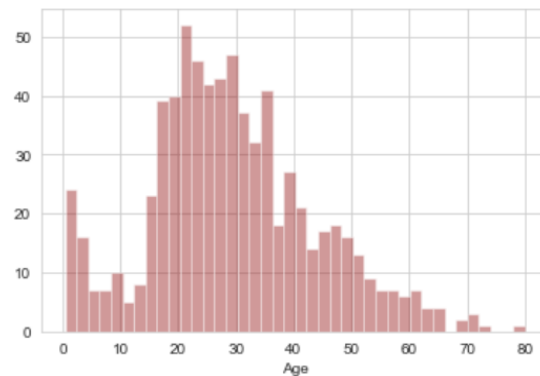
86 Pretty exiting!The passengers who died the most belonged to class three which is the lowest one.  
87 Another interesting observation is that nearly 60 per cents of class 2 passengers died , even third-  
88 class passengers survived more, although in proportion, of course, third-class passengers died  
89 more often.

## 90 What is the average age of Titanic passenger

91 Now I want to find out people of what age were on Titanic the most. Let's use function of seaborn  
92 that shows distribution of values.

```
93 1 sns.distplot(train['Age'].dropna(),kde=False,color='darkred',bins=40)
```

```
94 2 #Some explanations: I want transmit column 'Age' to sns but without NaN objects , to do it  
95     I need to type .dropna() after our data.The kda parameter is false cause I do not want  
96     to see kernel density estimation on our graph.
```

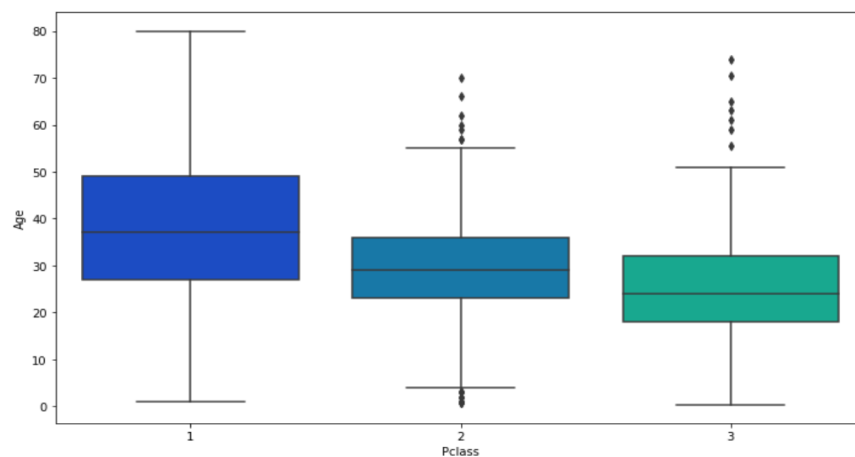


97  
98 As you can see the average age was around 20-30 .

## 99 Data cleaning

100 We want to fill in missing age data instead of just dropping the missing age data rows. One way to  
101 do this is by filling in the mean age of all the passengers. But this is a too wild way of imputation  
102 so we will use another method: we will understand what is the average age of each class and only  
103 after that we will fill in the missing data.

```
104 1 plt.figure(figsize=(12, 7))
105 2 sns.boxplot(x='Pclass',y='Age',data=train,palette='winter')
106 3 #Some explanations: we will use .boxplot , the x axis will be 3 our classes and the y axis
107 4 will be age.
```



108  
109 This boxplot give us a lot of information . These black lines on each box is the average value of  
110 this class. So depending on this information we will replace every NaN value in 'Age' column. Let's  
111 take 36 as average age of 1st class passenger , 29 as average age of 2nd class passenger and 24 as  
112 average age of 3rd class passenger.

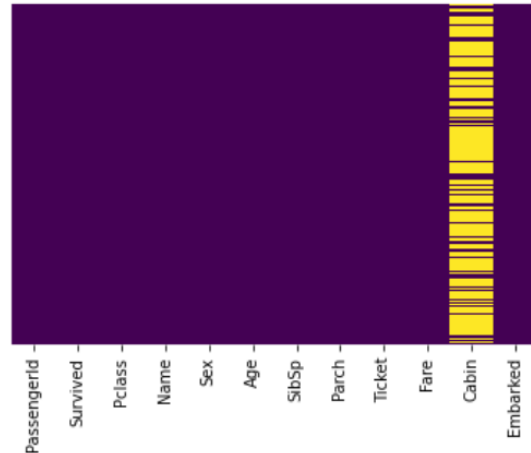
```
113 1 def impute_age(cols):
114 2     Age = cols[0]
115 3     Pclass = cols[1]
116 4
117 5     if pd.isnull(Age):
118 6         if Pclass == 1:
119 7             return 36
120 8         elif Pclass == 2:
121 9             return 29
12210         else:
12311             return 24
12412
12513     else:
12614         return Age
```

127 Now we will apply this function to our data set.

```
128 1 train['Age'] = train[['Age', 'Pclass']].apply(impute_age,axis=1)
129 2 #Some explanations: I used function impute-age by built-in function apply and transmit all
130 the necessary values.
```

131 Let's look at our heatmap of isNaN again.

```
132 1 sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```



133

134 Great! The last thing that we need to do is remove 'Cabin' column at all because there is too little  
135 information about it. Also I will show how our data now look like.

```
136 1 train.drop('Cabin',axis=1,inplace=True) #removing 'cabin'
137 2 train.head()
```

Pass-ID	Surv	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
2	1	1	Cumings, Mrs. John Bradley	female	38.0	1	0	PC 17599	71.2833	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
4	1	1	Futrelle, Mrs. Jacques Heath	female	35.0	1	0	113803	53.1000	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	S
6	0	3	Moran, Mr. James	male	24	0	0	330877	8.4583	Q
7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	S
8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	S
9	1	3	Johnson, Mrs. Oscar W	female	27.0	0	2	347742	11.1333	S
10	1	2	Nasser, Mrs. Nicholas	female	14.0	1	0	237736	30.0708	C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	S
12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	S

## 139 Conlusions and hypothesizing

140 We understood that:

141 I On Titanic, men died in the majority.

142 II Third-class passengers had almost no chance of survival.

143 III Young people were in greater danger than old people cause in most cases young people were  
144 3rd class passengers.

145 And after this thoughts I want to make a hypothesis:if you are a passenger of the Titanic and want  
146 to survive with the greatest probability, you should be the little daughter of very rich parents.

## 147 Hypothesis testing

148 We start this analysis by adding a new column to the 'train data frame'. Use the Survived column  
149 to map to the new column with factors 0 for 'no' and 1 for 'yes' using the map method

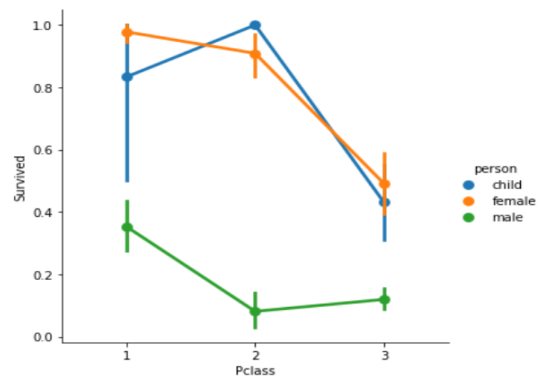
```
150 1 train['Survivor'] = train.Survived.map({0:'no', 1:'yes'})
```

151 Also let's add a 'Person' column which will contain three types : male , female or child.

```
152 1 # Function which will determine is this passenger a child
153 2 def whoIsPerson(passenger):
154 3     age, sex = passenger
155 4
156 5     if age < 16:
157 6         return 'child'
158 7     else:
159 8         return sex
160 9
161 10 train['person'] = train[['Age', 'Sex']].apply(whoIsPerson, axis=1)
```

163 Now let's see how graph with information about class/gender and survival looks like.

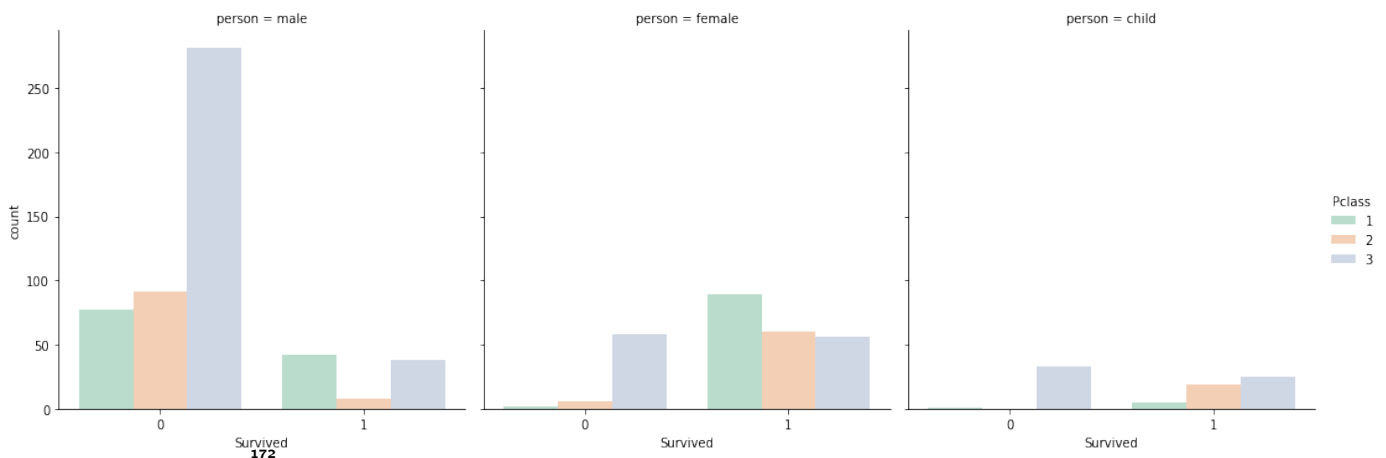
```
164 1 sns.factorplot('Pclass', 'Survived', hue='person', data=train, order=range(1,4),
165 2     hue_order = ['child', 'female', 'male'])
```



166

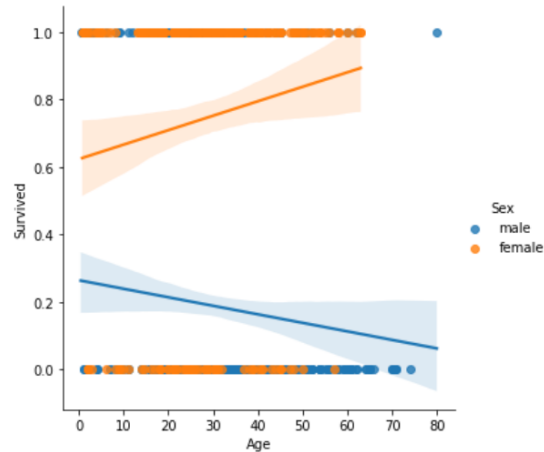
167 From the graph above, it is clear that being a man and even a third class greatly reduces the chances  
168 of survival

```
169 1 sns.factorplot('Survived', data=train, hue='Pclass', kind='count', palette='Pastel2',
170 2     hue_order=range(1,4),
171 3     col='person')
```

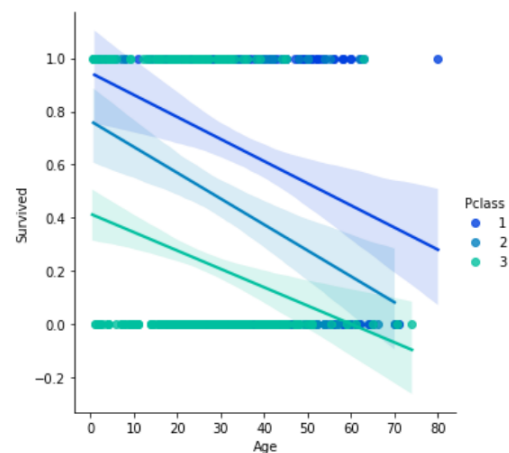


173 And last thing that we need to prove is correlation between class and survival ( as I said you  
174 should be daughter of RICH parents who would probably buy seats at first class)

```
175 1 sns.lmplot('Age', 'Survived', data=train, hue='Sex')  
176 2 sns.lmplot('Age', 'Survived', hue='Pclass', data=train, palette='winter', hue_order=range  
177 (1,4))
```



**Figure 1.** Age is X axis , Survived is Y axis grouped by sex



**Figure 2.** Age is X axis , Survived is Y axis grouped by class

178

179 As Graph 1 above showed : older women are more likely to survive than older men , but the second  
180 graph shows that the probability that a person will survive decreases with increasing age. I guess  
181 that all this information is enough to make a results review and the final conclusion.



## 182 Results , conclusion

183 I hypothesized about the survival factors on board the Titanic . Let me briefly remind you: *in*  
184 *order to survive, it is desirable to be a little girl of rich parents.* And as we learned and proved by  
185 statistics, children had more chances of survival than men, namely, *the chances of survival decrease*  
186 *with increasing age.*

187 Then we needed to confirm that women, on average, were saved more often than men in most  
188 cases. Graphs show that a woman of any age was saved more often than a man. It remains to  
189 prove the relationship between survival and class, but it was not difficult. As it turned out, almost  
190 all the passengers of the 3rd class could not escape, more than half of the second class were also  
191 unlucky, but the third class passengers showed the best "survival rates". So my hypothesis was  
192 confirmed, which is good news. However, in the course of work, I learned a lot of sad facts that we  
193 can prevent in the future.

194 The main conclusion that can be drawn based on my hypothesis is the following one: Passengers  
195 of any class, age or gender should have an equal chance of salvation.

196 The end

## Links:

1. Code on GitHub: <https://github.com/FyodoRaev/TitanicData>
2. EDA guides that I used : <https://youtu.be/-o3AxdVcUtQ>  
, [https://youtu.be/Ea\\_KAcdv1vs](https://youtu.be/Ea_KAcdv1vs)
3. Pandas library documentation : [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/missing\\_data.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html)
4. Seaborn library documentation : <https://seaborn.pydata.org/>