Exploratory Data Analysis of Titanic data set

Fyodor Raevskiy[1]

***For correspondence:**
xboxraevskii@mail.ru (FMS);
@iwarnedyouaboutstairss
(Telegram)

[†]Project for Tinkoff Generation
[‡]The data set was taken from kaggle.com

[1]January 2022

Abstract I am writing this report to analyze information about people who died and survived on Titanic, make some hypotheses on this topic and try to prove it by statistics.

Introduction

This is my first EDA so all calculations and conclusions will be done step by step. I used the Titanic Data Set from Kaggle. This is a very famous data set and it is frequently a student's first step in Exploratory Data Analysis and machine learning.

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

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Reading the data

One of the most important first steps is to understand what kind of data it is . 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 this case, we will use pandas.read-CSV() which is the most popular at newbies.

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

How our data looks like

The easiest way to look on the data set is to show just a few of the first lines of the table.To begin with, let's print first 12 strings.

```
train.head()
```

What does each column mean?

After seeing all the information, the reader will need some explanations.

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

There is a column which is called parch. It shows number of brothers, sisters, stepbrothers, step-sisters, spouses on the board of Titanic.

'Embarked' shows port of embarkation.'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	Embark
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

Missing data

According to the table 1, there are a lot of missing points about the cabin and the age of a passenger. Moreover, I would like to remove them and I will explain why. Getting rid of NaN objects in most cases caused by simplicity. As data comes in many shapes and forms, we aim to find the easiest way of understanding statistics. We can use seaborn to create a simple heat map to see where the information is missing.

```
train.isnull()
```

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

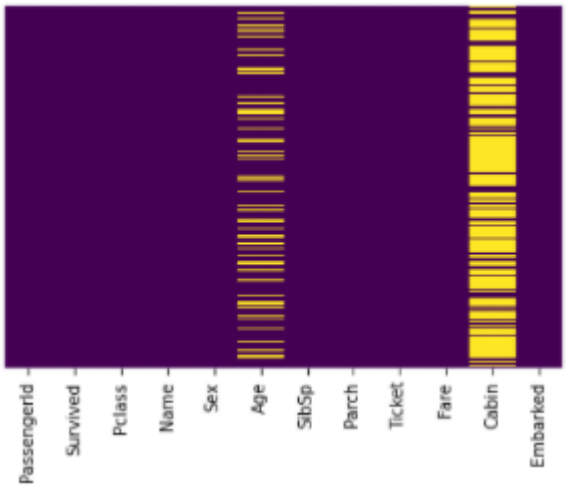
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3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

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

However, this is not the best way to remove missing data because it becomes more complicated , if there is a lot of information. So, I suggest using method of visualisation . Let 's create a graph that will show which column has the most of the missing data.

```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
#Some explanations: when I typed train.isnull() in brackets I meant  that it should take
    train.isnull() and whenever it's true, sbs will display it in yellow color on a graph.
     The y axis is all passengers but I typed yticklabels=False because I do not want to
    see all of them.And the last two arguments are for graph's appearance.
```

isNull graphic:



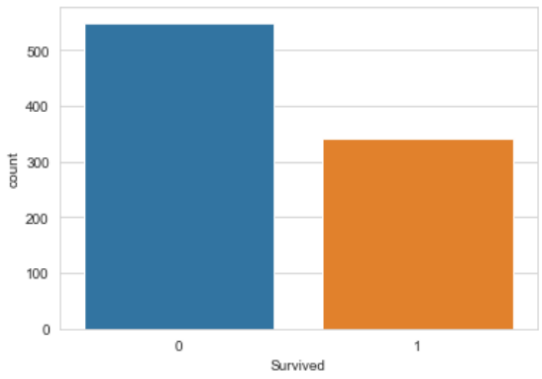
What is supposed to be done?

Therefore, almost 20 percent of the age of passengers is missing. But it seems to me that such a proportion is reasonable enough to replace this data with something sensible. But there is a lack of information about the cabins, and I will most likely get rid of this column or replace it with "Is there information about the cabin: 1 for yes and 0 for no".

Let's continue visualising some more of the data.

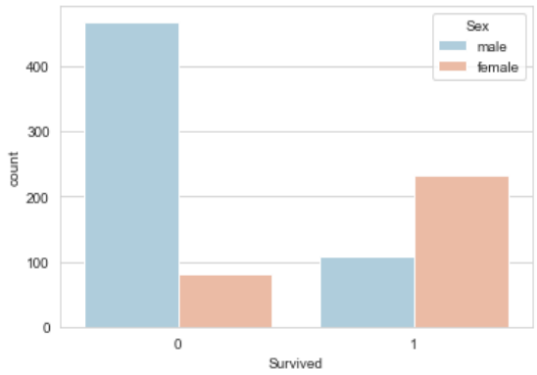
Let's find out how many people survived on Titanic

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



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

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

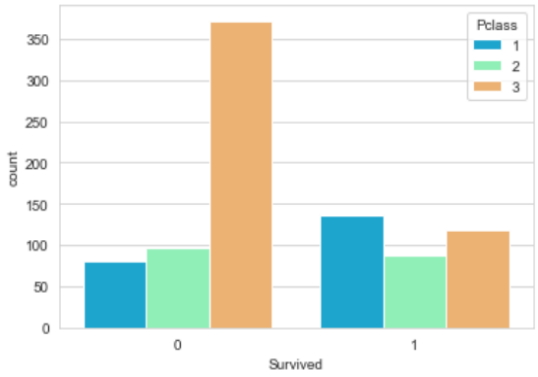


Most of the men and at least 80% of the women died.

Comparing different classes

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

```
sns.countplot(x='Survived',hue='Pclass',data=train,palette='rainbow')
#Some explanations: now I typed to sbs that it must take column 'Survived' and count,
     depending on column 'Pclass', how many passengers of each class have survived.
```

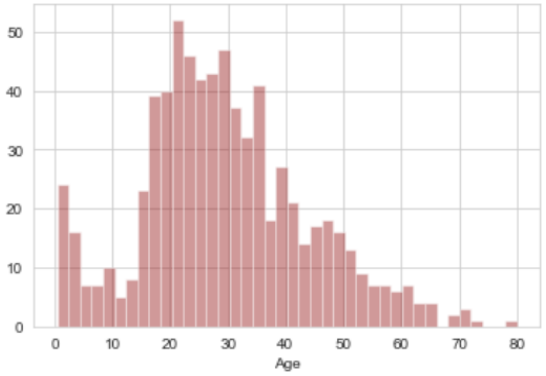


The passengers who died the most belonged to class 3 which is the lowest one. Another interesting observation is that nearly 60 per cents of class 2 passengers died , even third-class passengers survived more, although in proportion third-class passengers died more often.

What is the average age of Titanic passenger

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

```
sns.distplot(train['Age'].dropna(),kde=False,color='darkred',bins=40)
#Some explanations: I want transmit column 'Age' to sns but without NaN objects , to do it
     I need to type .dropna() after our data.The kda parameter is false cause I do not want
      to see kernel density estimation on our graph.
```

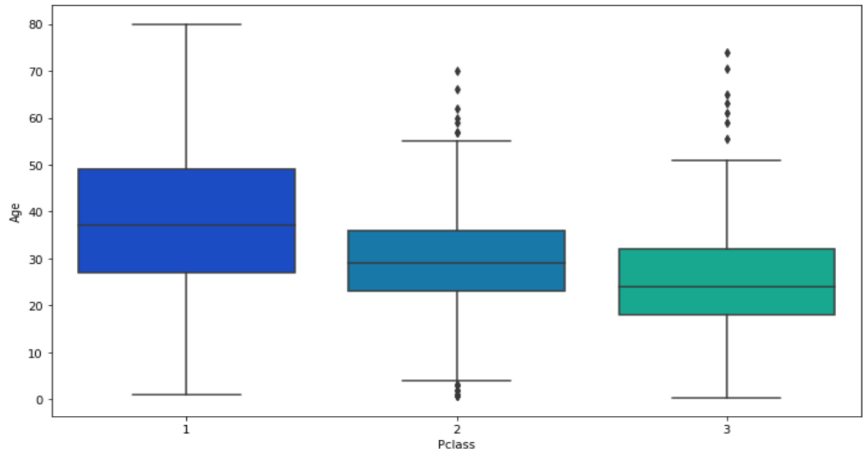


As you can see the average age was around 20-30 .

Data cleaning

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

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



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

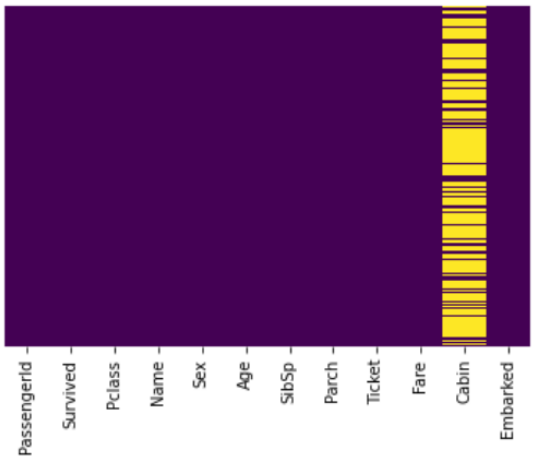
```
def impute_age(cols):
    Age = cols[0]
    Pclass = cols[1]

    if pd.isnull(Age):
        if Pclass == 1:
            return 36
        elif Pclass == 2:
            return 29
        else:
            return 24

    else:
        return Age
```

Now, it is time to apply this function to data set.

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

Let's look at our heatmap of isNaN again.

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



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

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

Conlusions and hypothesizing

We understood that:

I The vast majority of the passengers who died on Titanic were men.

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

III Young people were in greater danger than older people as in most cases young people were 3rd class passengers.

After these thoughts I would like to make a hypothesis.Most of the children and elderly women were among the survivors. It could mean that human that human morality contributed to the salvation of these people .Perhaps, quite others would have survived, if not for the nobility of people of the 3rd class.

Hypothesis testing

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

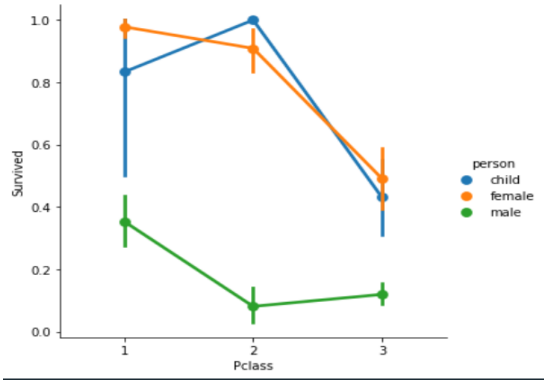
```
train['Survivor'] = train.Survived.map({0:'no', 1:'yes'})
```

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

```
# Function which will determine is this passenger a child
def whoIsPerson(passenger):
    age, sex = passenger

    if age < 16:
        return 'child'
    else:
        return sex


train['person'] = train[['Age', 'Sex']].apply(whoIsPerson, axis=1)
```

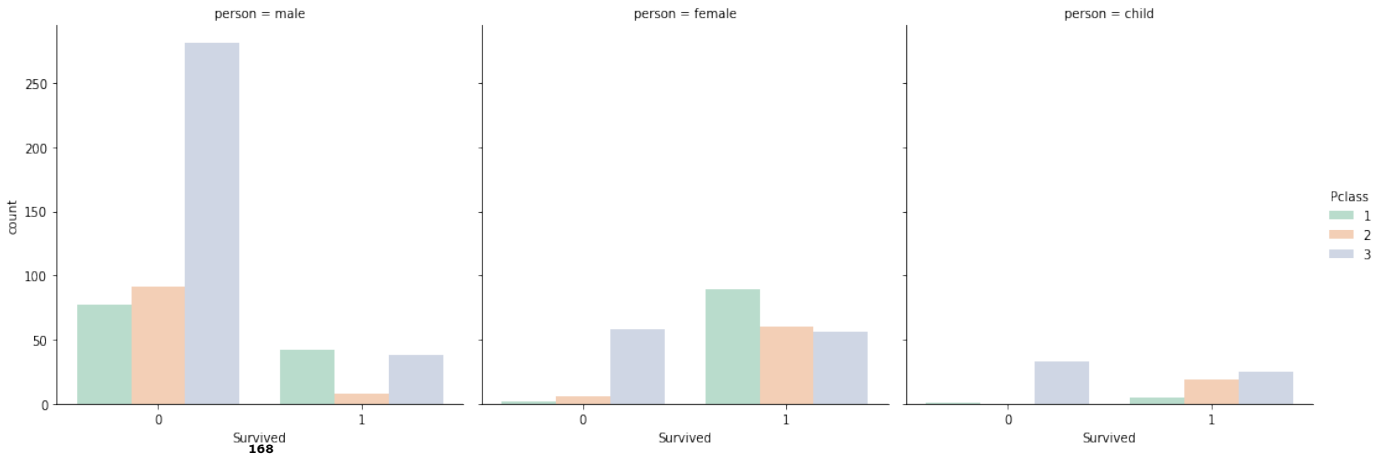
Now let's see how graph with information about class/gender and survival looks like.

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



The graph above confirms the assumption that men of the 3rd class survived much less often than passengers of the 1st class.

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



Now I will show the correlation between the survival rate of a passenger and his age, gender and class

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

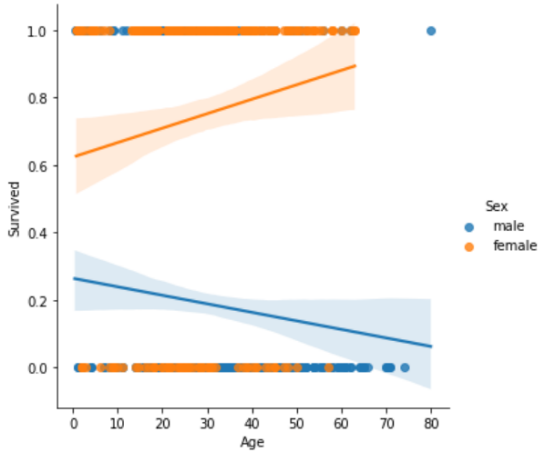


Figure 1. Age is X axis , Survived is Y axis grouped by sex

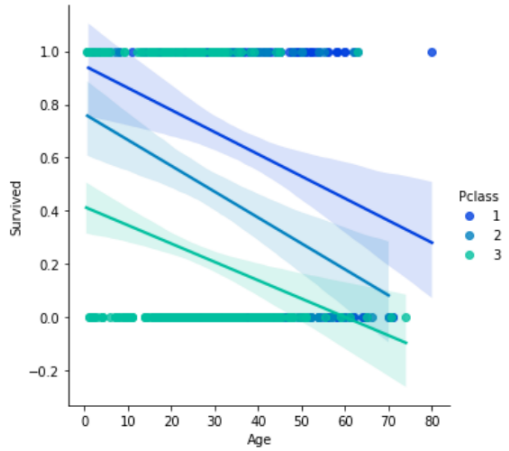


Figure 2. Age is X axis , Survived is Y axis grouped by class

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

Results , conclusion

To conclude, I assumed that it was morality that prevented more of the 3rd class from surviving. My arguments are fully supported by the data because we found out that elderly women and children were rescued much more often. How important is this fact? Today, predictions and instructions for rescue services must view morality as a very important factor in human behaviour. For example, when designing features or machine learning algorithms, our goal is usually to isolate the factors of variation that explain the observed data. Accordingly, my hypothesis can help analysts of rescue services in preventing dangerous events, which, in my opinion, is important in the modern world.

The end

Links:

1. Code on GitHub: https://github.com/FyodoRaev/TitanicDat
2. Pandas library documentation : https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html
3. Seaborn library documentation : https://seaborn.pydata.org/