Practice-PandasTitanic

October 7, 2024

1 Name: Ramya Chowdary Patchala

2 Data Analysis with Pandas - PRACTICE 1

We will perform a data analysis on the **RMS Titanic** passenger list. The RMS Titanic is one of the most famous ocean liners in history. On April 15, 1912 it sank after colliding with an iceberg in the North Atlantic Ocean. To learn more, read here: https://en.wikipedia.org/wiki/RMS_Titanic

Our goal today is to perform a data analysis on a subset of the passenger list. We're looking for insights as to which types of passengers did and didn't survive. Women? Children? 1st Class Passengers? 3rd class? Etc.

I'm sure you've heard the expression often said during emergencies: "Women and Children first" Let's explore this data set and find out if that's true!

Before we begin you should read up on what each of the columns mean in the data dictionary. You can find this information on this page: https://www.kaggle.com/c/titanic/data

2.1 Loading the data set

First we load the dataset into a Pandas `DataFrame` variable. The `sample(10)` method takes a random sample of 10 passengers from the data set.

[1]:
```python
import pandas as pd
import numpy as np

# this turns off warning messages
import warnings
warnings.filterwarnings('ignore')

passengers = pd.read_csv('https://raw.githubusercontent.com/mafudge/datasets/
↪master/ist256/12-pandas/titanic.csv')
passengers.sample(10)
```

[1]:
```
     PassengerId  Survived  Pclass                              Name  \
340          341         1       2     Navratil, Master. Edmond Roger
306          307         1       1            Fleming, Miss. Margaret
153          154         0       3      van Billiard, Mr. Austin Blyler
853          854         1       1          Lines, Miss. Mary Conover
622          623         1       3                   Nakid, Mr. Sahid
```

```
6              7         0      1                     McCarthy, Mr. Timothy J
82            83         1      3              McDermott, Miss. Brigdet Delia
775          776         0      3  Myhrman, Mr. Pehr Fabian Oliver Malkolm
816          817         0      3                Heininen, Miss. Wendla Maria
317          318         0      2                        Moraweck, Dr. Ernest

        Sex   Age  SibSp  Parch             Ticket      Fare Cabin Embarked
340    male   2.0      1      1             230080   26.0000    F2        S
306  female   NaN      0      0              17421  110.8833   NaN        C
153    male  40.5      0      2          A/5. 851   14.5000   NaN        S
853  female  16.0      0      1           PC 17592   39.4000   D28        S
622    male  20.0      1      1               2653   15.7417   NaN        C
6      male  54.0      0      0              17463   51.8625   E46        S
82   female   NaN      0      0             330932    7.7875   NaN        Q
775    male  18.0      0      0             347078    7.7500   NaN        S
816  female  23.0      0      0  STON/O2. 3101290    7.9250   NaN        S
317    male  54.0      0      0              29011   14.0000   NaN        S
```

2.2 How many survived?

One of the first things we should do is figure out how many of the passengers in this data set survived. Let's start with isolating just the `'Survived'` column into a series:

[2]:
```
passengers
```

[2]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           …        …      …
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                               Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                             Allen, Mr. William Henry    male  35.0      0
..                                                 …     …   …    …
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
```

```
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                                Behr, Mr. Karl Howell    male  26.0      0
890                                  Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …       …    …      …
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

[3]:
```python
passengers['Survived'].sample(10)
```

[3]:
```
631    0
34     0
343    0
869    1
870    0
452    0
459    0
471    0
377    0
554    1
Name: Survived, dtype: int64
```

There's too many to display so we just display a random sample of 10 passengers.

- 1 means the passenger survivied
- 0 means the passenger died

What we really want is to count the number of survivors and deaths. We do this by querying the `value_counts()` of the `['Survived']` column, which returns a `Series` of counts, like this:

[4]:
```python
passengers['Survived'].value_counts()
```

[4]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```

Only 342 passengers survived, and 549 perished. Let's observe this same data as percentages of the whole. We do this by adding the `normalize=True` named argument to the `value_counts()` method.

[5]:
```
passengers['Survived'].value_counts(normalize=True)
```

[5]:
```
Survived
0    0.616162
1    0.383838
Name: proportion, dtype: float64
```

Just 38% of passengers in this dataset survived.

2.2.1 1.1 You Code

FIRST Write a Pandas expression to display counts of males and female passengers using the `Sex` variable:

[6]:
```
# todo write code here
passengers["Sex"].value_counts()
```

[6]:
```
Sex
male      577
female    314
Name: count, dtype: int64
```

2.2.2 1.2 You Code

NEXT Write a Pandas expression to display male /female passenger counts as a percentage of the whole number of passengers in the data set.

[13]:
```
# todo write code here
passengers["Sex"].value_counts(normalize=True)
```

[13]:
```
Sex
male      0.647587
female    0.352413
Name: proportion, dtype: float64
```

If you got things working, you now know that **35% of passengers were female**.

2.3 Who survivies? Men or Women?

We now know that 35% of the passengers were female, and 65% we male.

The next thing to think about is how do survivial rates affect these numbers?

If the ratio is about the same for surviviors only, then we can conclude that your **Sex** did not play a role in your survival on the RMS Titanic.

Let's find out.

[14]:
```
survivors = passengers[passengers['Survived'] ==1]
survivors['PassengerId'].count()
```

[14]:
```
342
```

Still **342** like we discovered originally. Now let's check the **Sex** split among survivors only:

[15]:
```
survivors['Sex'].value_counts()
```

[15]:
```
Sex
female    233
male      109
Name: count, dtype: int64
```

WOW! That is a huge difference! But you probably can't see it easily. Let's represent it in a `DataFrame`, so that it's easier to visualize:

[16]:
```
sex_all_series = passengers['Sex'].value_counts()
sex_survivor_series = survivors['Sex'].value_counts()

sex_comparision_df = pd.DataFrame({ 'AllPassengers' : sex_all_series,␣
↪'Survivors' : sex_survivor_series })
sex_comparision_df['SexSurvivialRate'] = sex_comparision_df['Survivors'] /␣
↪sex_comparision_df['AllPassengers']
sex_comparision_df
```

[16]:

Sex	AllPassengers	Survivors	SexSurvivialRate
female	314	233	0.742038
male	577	109	0.188908

[17]:
```
sex_all_series = passengers['Sex'].value_counts()
sex_all_series
```

[17]:
```
Sex
male      577
female    314
Name: count, dtype: int64
```

So, females had a 74% survival rate. Much better than the overall rate of 38%

We should probably briefly explain the code above.

- The first two lines get a series count of all passengers by Sex (male / female) and count of survivors by sex
- The third line creates a Pandas DataFrame. Recall a pandas dataframe is just a dictionary of series. We have two keys 'AllPassengers' and 'Survivors'
- The fourth line creates a new column in the dataframe which is just the survivors / all passengers to get the rate of survival for that Sex.

2.4 Feature Engineering: Adults and Children

Sometimes the variable we want to analyze is not readily available, but can be created from existing data. This is commonly referred to as **feature engineering**. The name comes from machine learning where we use data called *features* to predict an outcome.

Let's create a new feature called `'AgeCat'` as follows:

- When **Age** <=18 then 'Child'
- When **Age** >18 then 'Adult'

This is easy to do in pandas. First we create the column and set all values to `np.nan` which means 'Not a number'. This is Pandas way of saying no value. Then we set the values based on the rules we set for the feature.

[18]:
```
passengers['AgeCat'] = np.nan # Not a number
passengers['AgeCat'][ passengers['Age'] <=18 ] = 'Child'
passengers['AgeCat'][ passengers['Age'] > 18 ] = 'Adult'
passengers.sample(5)
```

[18]:
```
     PassengerId  Survived  Pclass                             Name     Sex  \
701          702         1       1  Silverthorne, Mr. Spencer Victor    male
590          591         0       3              Rintamaki, Mr. Matti    male
750          751         1       2                 Wells, Miss. Joan  female
598          599         0       3                 Boulos, Mr. Hanna    male
61            62         1       1               Icard, Miss. Amelie  female

      Age  SibSp  Parch             Ticket     Fare Cabin Embarked AgeCat
701  35.0      0      0           PC 17475  26.2875   E24        S  Adult
590  35.0      0      0  STON/O 2. 3101273   7.1250   NaN        S  Adult
750   4.0      1      1              29103  23.0000   NaN        S  Child
598   NaN      0      0               2664   7.2250   NaN        C    NaN
61   38.0      0      0             113572  80.0000   B28      NaN  Adult
```

Let's get the count and distrubutions of Adults and Children on the passenger list.

[19]:
```
passengers
```

[19]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           …         …       …
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3
```

```
                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                               Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                             Allen, Mr. William Henry    male  35.0      0
..                                                 …     …    …    …
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked AgeCat
0        0         A/5 21171   7.2500   NaN        S  Adult
1        0          PC 17599  71.2833   C85        C  Adult
2        0  STON/O2. 3101282   7.9250   NaN        S  Adult
3        0            113803  53.1000  C123        S  Adult
4        0            373450   8.0500   NaN        S  Adult
..     …               …      …   …      …      …
886      0            211536  13.0000   NaN        S  Adult
887      0            112053  30.0000   B42        S  Adult
888      2        W./C. 6607  23.4500   NaN        S    NaN
889      0            111369  30.0000  C148        C  Adult
890      0            370376   7.7500   NaN        Q  Adult

[891 rows x 13 columns]
```

And here's the percentage as a whole:

[20]:
```
passengers['AgeCat'].value_counts(normalize=True)
```

[20]:
```
AgeCat
Adult    0.805322
Child    0.194678
Name: proportion, dtype: float64
```

So close to **80%** of the passengers were adults. Once again let's look at the ratio of `AgeCat` for survivors only. If your age has no bearing of survivial, then the rates should be the same.

Here are the counts of Adult / Children among the survivors only:

[21]:
```
survivors = passengers[passengers['Survived'] ==1]
survivors['AgeCat'].value_counts()
```

[21]:
```
AgeCat
Adult    220
```

```
Child     70
Name: count, dtype: int64
```

2.4.1 1.3 You Code

Calculate the `AgeCat` survival rate, similar to how we did for the `SexSurvivalRate`.

[23]:
```
agecat_all_series = passengers['AgeCat'].value_counts()
agecat_survivor_series = survivors['AgeCat'].value_counts()

# todo make a data frame, add AgeCatSurvivialRate column, display dataframe
age_comparison_df = pd.DataFrame({'All': agecat_all_series,'Survivors':
 ↪agecat_survivor_series })
age_comparison_df['AgeSurvivalRate'] = age_comparison_df['Survivors']/
 ↪age_comparison_df['All']
age_comparison_df
```

[23]:

AgeCat	All	Survivors	AgeSurvivalRate
Adult	575	220	0.382609
Child	139	70	0.503597

So, children had a 50% survival rate, better than the overall rate of 38%

2.5 So, women and children first?

It looks like the RMS really did have the motto: "Women and Children First."

Here are our insights. We know:

- If you were a passenger, you had a 38% chance of survival.
- If you were a female passenger, you had a 74% chance of survival.
- If you were a child passenger, you had a 50% chance of survival.

2.5.1 Now you try it for Passenger Class

Repeat this process for `Pclass` The passenger class variable. Display the survival rates for each passenger class. What does the information tell you about passenger class and survival rates?

I'll give you a hint… "Class matters!"

2.5.2 1.4 You Code

[24]:
```
# todo: repeat the analysis in the previous cell for Pclass
all_pclass_series= passengers['Pclass'].value_counts()
survived_pclass_series = passengers[ passengers['Survived'] == 1]['Pclass'].
 ↪value_counts()
pclass_df = pd.DataFrame( { 'All' : all_pclass_series, 'Survived' :␣
 ↪survived_pclass_series})
pclass_df['Ratio'] = pclass_df['Survived'] /pclass_df['All']
```

```
pclass_df
```

[24]:
```
        All  Survived     Ratio
Pclass
1       216       136  0.629630
2       184        87  0.472826
3       491       119  0.242363
```

[25]:
```
passengers[ passengers['Survived'] == 1]['Pclass'].value_counts()
```

[25]:
```
Pclass
1    136
3    119
2     87
Name: count, dtype: int64
```

Not a big surprise. The 1st class passengers had a 62.9% survival rate!

2.6 What have we learned?

Your best odds of survival were:

- First class ticket `Pclass=1`
- Female
- Child

Your job is to check the survival rate of those individuals. Here's the process

1. filter the passengers data frame by the above criteria
2. normalize the value counts of survived.

Learn that while only 38% of all passengers survivied, 90.9% passengers meeting this criteria survivied!

2.6.1 1.5 You Code

[26]:
```
# TODO
ffc_df = passengers[passengers['Pclass'] == 1][passengers['Sex']==
 ↪'female'][passengers['AgeCat']=='Child']
ffc_df['Survived'].value_counts(normalize = True)
```

[26]:
```
Survived
1    0.909091
0    0.090909
Name: proportion, dtype: float64
```

3 Metacognition

3.0.1 Rate your comfort level with this week's material so far.

1 ==> I don't understand this at all yet and need extra help. If you choose this please try to articulate that which you do not understand to the best of your ability in the questions and comments section below.
2 ==> I can do this with help or guidance from other people or resources. If you choose this level, please indicate HOW this person helped you in the questions and comments section below.
3 ==> I can do this on my own without any help.
4 ==> I can do this on my own and can explain/teach how to do it to others.

`--== Double-Click Here then Enter a Number 1 through 4 Below This Line ==--`

3.0.2 4