Fall -23 DATA -220

Mathematical Method
for Data Analytics

HW - 1

Name :- Prayag Nikul Purani
SJSU ID :- 017416737

Problem 1 :-

Given data :

One-year-old flounder

mean = 126 mm

σ = 18 mm

Two-year-old-flounder

mean = 162 mm

σ = 28 mm

a) Z-score for one year old flounder of $x = 155$ mm

$$\Rightarrow Z_1 = \frac{x - \mu}{\sigma}$$

$$= \frac{155 - 126}{18}$$

$$= \frac{29}{18} = 1.6111 \text{ mm}$$

$$Z_1 = 1.6111 \text{ mm}$$

b) Z-score for two-year-old flounder with length (x) = 185 mm.

$$\Rightarrow Z_2 = \frac{x - \mu}{\sigma}$$

$$= \frac{185 - 162}{28}$$

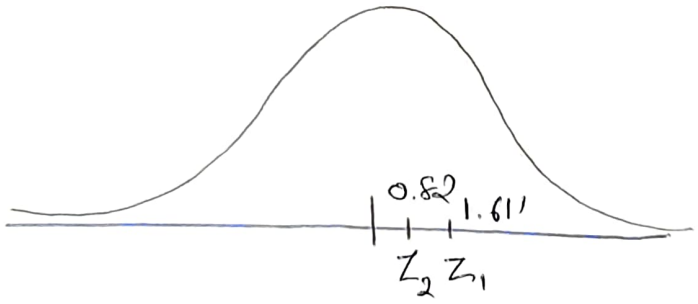
$$= \frac{23}{28} = 0.8214 \text{ mm}$$

$$Z_2 = 0.8214 \text{ mm}$$

c)

after normalization

$$Z_1 > Z_2$$



So from the graph of the value we can see $Z_1 > Z_2$

So, the lengths of 1-year-flounder is greater than length of 2-year flounder.

Problem 2:

Given data :-

intercept = c = 160.19

coefficient = m = 0.10

a) Regression equation

$$\hat{y} = mx + c$$

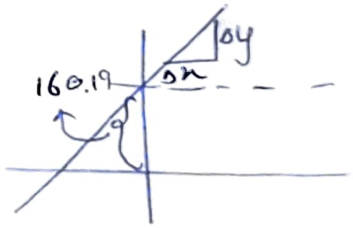
from the given data.

$$\hat{y} = 0.10x + c$$

$$\boxed{\hat{y} = 0.10x + 160.19}$$

b) Slope is 0.10

$$\frac{\Delta y}{\Delta x} = \frac{0.10}{1}$$



So, every run of x by 1 there is rise of y by 0.10

c) Size (square feet) = 2800 sq feet = x

Substitution in Regression eq^n.

(predicted SP) $\hat{y} = 0.1(2800) + 160.19$

$\boxed{\hat{y} = 440.19}$ selling price ($1000s)

d) Size (sq feet) = 3049 feet²

$$\hat{y} = mx + c$$

$$\hat{y} = 0.1(3049) + 160.19$$

$$\boxed{\hat{y} = 465.09}$$ selling prize (1000s $)

as the data w.r.t to 3049 is given a 475

so we can get error/residual by

= [actual (y) − predicted ($\hat{y}$)]

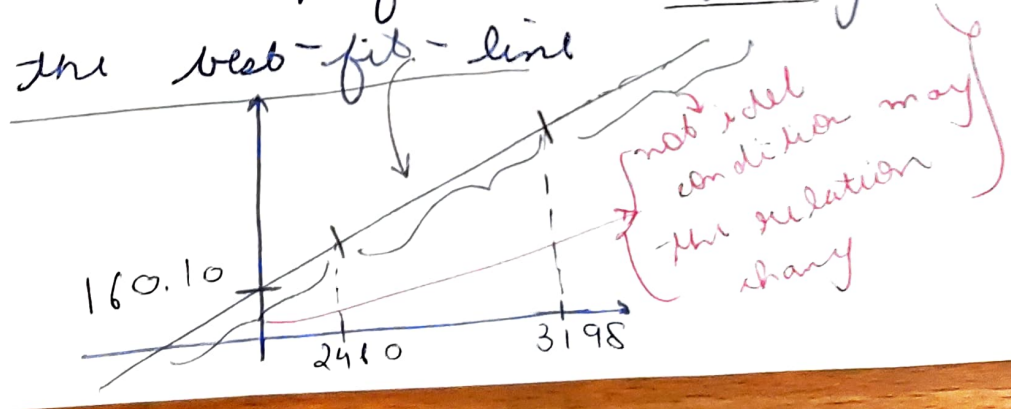
= [475 − 465.09]

$$\boxed{\text{error/residual} = 9.91 \text{ selling price (\$ 1000s)}}$$

e) Size (sq feet) = 5000 sq feet

so min size = 2460 sq feet
max size = 3198 sq feet

so x = 5000 sq feet is not belong to the best-fit-line



b) Size (sq feet) = 2555 sq feet

$$\hat{y} = 0.1(2555) + 160.19$$

$$\boxed{\hat{y} = 415.69 \text{ selling price (\$1000s)}}$$

error/residual = [actual (y) − predicted ($\hat{y}$)]

= [426 − 415.69]

$$\boxed{\text{error/residual} = 10.31 \text{ selling price (\$1000s)}}$$

Problem 3:-

Gender Owner	Cats	Dogs	Other pets	total
Female	100	50	30	180
Male	50	50	20	120
	150	100	50	300

a) $P(\text{cats}) = \frac{\text{fav. outcome}}{\text{total outcome}} = \frac{150}{300} = \frac{1}{2} = 0.5$

b) $P(\text{dog} \cup \text{female}) = P(\text{dog}) + P(\text{female}) - P(\text{dog} \cap \text{female})$

$$= \frac{100}{300} + \frac{180}{300} - \frac{50}{300}$$

$$= \frac{230}{300} = 0.7666$$

c) P(other pet ∩ female) =

fav of outcome / total outcomes : $\frac{30}{300} = 0.1$

d) P(cat or dog) = P(cat) + P(dog) − P(cat ∩ dog)

P(cat ∩ dog) = 0 [as it mutually exclusive events]

$= \frac{150}{300} + \frac{100}{300}$

$= \frac{250}{300} = \frac{25}{30} = 0.8333$

Problem 4

P(D) = 0.02

P(S) = 0.98

a) D < D, S

S < D, S

Sample { DD, DS, SD, SS }

2 book so $2^2 = 4$ sample space

$P(DD) = P(D) \times P(D)$

as the selecting of book 2 is not depending on the selection of 1st book so are mutually exclusive event

$P(DD) = 0.02 \times 0.02$

$\boxed{P(DD) = 0.0004}$

b) 4 book are randomly selected

so sample space $= 2^4 = 2 \times 2 \times 2 \times 2$
$= 16$ element.

$P(DSSS)$ => as the events are mutually exclusive

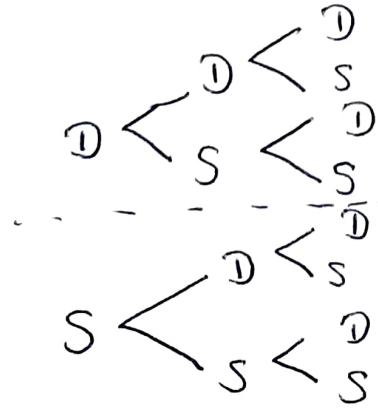
so,

$P(DSSS) = P(D) * P(S) * P(S) * P(S)$

$= 0.02 * 0.98 * 0.98 * 0.98$

$\boxed{P(DSSS) = 0.0188238}$

c) 3 book are randomly selected.



Sample space { DDD, DDS, DSD, DSS SDD, SDS, SSD, SSS },

So probability of at least one defective = $1 - P(SSS)$

$= 1 - P(S) * P(S) * P(S)$

as they are mutually exclusive

$= 1 - [0.98]^3$

probability = 0.058808
at least defective

d) So the sample space of 5 books selected will $2^5 = 32$.

not fisible to write but the probability of at least one defective will be

$= 1 - P(S) * P(S) * P(S) * P(S) * P(S)$

$= 1 - [P(S)^5]$

$= 1 - (0.98)^5$

probability of at least one defective = 0.09607

Fall 2023 DATA 220 Mathematical Methods for Data Analytics

Homework – 1

Name :- Prayag Nikul Purani

SJSU Id :- 017416737

Problem 5 - Data Analysis using Titanic Dataset (Coding):-

The important step for the rest of the homework is the importing the data so it can be done using pandas.

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns

In [2]: df = pd.read_csv('titanic.csv')

In [3]: df.head()
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Fig 1 :- Importing csv using pandas

Question 1:

So, bar plot is the best representation for the visualization of the different entities which provides the knowledge of the data in quick access and the users or the viewer can easily get what is the relation between values and can get the idea of the graph or the distribution of data is going to be

a. First bar chat shows the distribution of gender, so on x-axis it will have the unique values in the column which is male and female and for y-axis it will have the counts. This bar chart will tell us how many numbers of male and female where there in the database.

Question 1: Bar Plots

```
In [181]: #### (A)
          # Claculating the numbers of males and females for y-axis
          counts = list(df['Sex'].value_counts())
          # Calculating the unique values from the column "Sex" for x-axis.
          values = list(df['Sex'].unique())

In [182]: # Using mathplot lib to make bar graph
          plt.bar(values,counts)
          plt.xlabel("Sex")
          plt.ylabel("Counts")
          plt.title("Bar Plot for Sex Vs Count")
          plt.show

Out[182]: <function matplotlib.pyplot.show(close=None, block=None)>
```

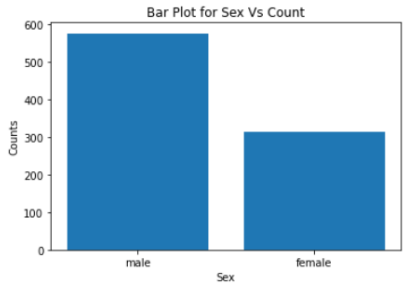


Fig 2 :- Bar chart for distribution of gender

b. Second bar chart show the number of passengers boarded from each stop with the ship starting point as Southampton and so on. So, the graph will have the stops on the x-axis and the count of people on y-axis.

```
In [6]: #### (B)
        # Claculating the numbers of passenger boarded form each stop for y-axis
        counts = list(df['Embarked'].value_counts())
        # Calculating the unique values Embarked for x-axis.
        values = list(df['Embarked'].dropna().unique())

In [7]: # Using mathplot lib to make bar graph
        plt.bar(values,counts)
        plt.xlabel("Embarked")
        plt.ylabel("Counts")
        plt.title("Bar Plot for Embarked from different stops")
        plt.show

Out[7]: <function matplotlib.pyplot.show(close=None, block=None)>
```

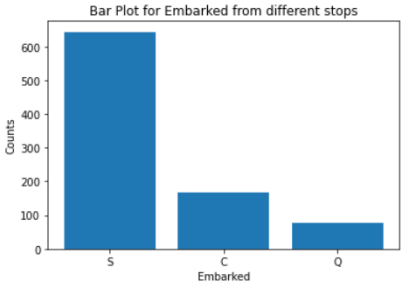


Fig 3 :- Bar chart for the distribution of Embarked from different stops

Question 2:-

The main aim of the question is to remove the outlines form the data so they will not affect the calculation and out dataset is reduced. Outlier detection is a process of identifying the data points in a dataset that lie far away from the majority of the data. These data points can either be significantly higher or lower than the other observations and can impact the overall results of the analysis. Second thing to do in this data set is

to remove null values to make the data-frame more cleaned and can be used for better visualization with efficient reading and values inputs for better predication.

a. The "Age" variable has missing values. Based on distribution (symmetric / non - symmetric), perform an appropriate missing value imputation technique to fill in the missing values. So, to know what's the pattern for the age column we need to make a histogram and then we if the graph is symmetric then we have to use mean imputation and if the graph is skewed right or left then we need to use median imputation method to fill the null values in the age column.

Question 2 (a):

```
In [8]: #age = list(df["Age"])
        ax = sns.distplot(df["Age"])
        plt.show()
```

```
C:\Users\Prayag Purani\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fun
ction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

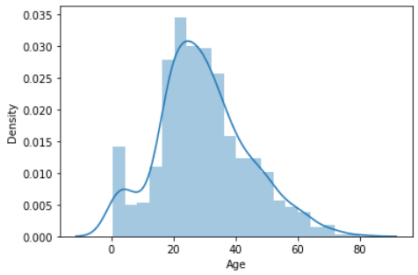


Fig 4 :- Histogram for the visualization of graph pattern

So, from the histogram we can see that the data is right skewed so we have to apply median imputation technique to fill the missing values.

```
In [9]: # the graph is right skewed so we will be using median imputation
        df["Age"] = df["Age"].fillna(df["Age"].median())

In [10]: df["Age"].isnull().unique()

Out[10]: array([False])
```

Fig 5 :- Meian imputation

So, after filling the age's nan with the median we can check that if it still has any nan left so the second line of code does the same thing and the result says the unique value in Age after Boolean function of isnull() is only false this means no null values are left in the data.

b. Now the detection of outliners can be done by three different methods and we can't use any inbuilt libraries of python.

(i) z-score:-

We are going to apply this formula $z = (x-\mu)/\sigma$ to find z-score in of the age column. The threshold specified in the question is between 3 to -3. So, after calculation we will print the index of outliners so to understand the number of outliners present in the data set according to the Z score method.

Question 2(b):

In [11]:
```
################################################## (i) ##################################################
# Calculating the Z-score for each age points
m = df["Age"].mean()
s = df["Age"].std()
z_score = list()
outliner_index = list()
threshold = 3
index = 0
for x in df["Age"]:
    z = (x-m)/s
    z_score.append(z)
    if(abs(z)>3):
        outliner_index.append(index)
    index = index + 1

df["zscore"] = z_score
```

In [12]:
```
#The index of the rows who are outliners by using the threshold 3 to -3 by using z-score
outliner_index
```

Out[12]:
```
[96, 116, 493, 630, 672, 745, 851]
```

Fig 6 :- Z-score

(ii) Modified z-score :-

So, the formula which we will be using is $0.6745(x_i - \tilde{x}) / MAD$

In [14]:
```
########################################################## (ii) ##################################################
#Claculating the modified z score
# MAD calculation
md = df["Age"].median()
l1 = list()
for x in df["Age"]:
    l1.append(abs(x-md))
l1.sort()
mid = len(l1) // 2
MAD = (l1[mid] + l1[~mid]) / 2
MAD
```

Out[14]:
```
6.0
```

In [15]:
```
# final modified z score calculation
modified_z_score = list()
outliner_index_2 = list()
threshold = 3
index = 0
for x in df["Age"]:
    z = 0.6745*(x-md)/MAD
    modified_z_score.append(z)
    if(abs(z)>3):
        outliner_index_2.append(index)
    index = index + 1

df["modified_zscore"] = modified_z_score
```

In [16]:
```
#The index of the rows who are outliners by using the threshold 3 to -3 by using modified z-score
print(*outliner_index_2)
```

```
11 15 33 54 78 94 96 116 152 164 170 172 174 183 195 232 252 268 275 280 305 326 366 381 386 438 456 467 469 483 487 492 493 54
5 555 570 587 625 626 630 644 647 659 672 684 694 745 755 772 788 803 827 829 831 851 879
```

Fig 6 :- Modified Z-score

(iii) Inter-Quartile range (IQR):-

The interquartile range (IQR) is a measure of statistical dispersion, or the spread of data. It is the range of values within which the middle 50% of scores reside. The IQR is defined as the difference between the 75th and 25th percentiles of the data.

In [17]:
```
############################################################ (iii) ##########################################################
# Identify outliers using IQR method
Q1 = df['Age'].quantile(0.25)
Q3 = df['Age'].quantile(0.75)
IQR = Q3 - Q1

# Define upper and lower bounds for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Create cleaned_data by removing outliers
df2 = df[(df['Age'] >= lower_bound) & (df['Age'] <= upper_bound)]
df2
```

Out[17]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	zscore	modified_zscore
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	-0.565419	-0.674500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	0.663488	1.124167
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	-0.258192	-0.224833
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	0.433068	0.786917
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	0.433068	0.786917
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S	-0.181385	-0.112417
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S	-0.795839	-1.011750
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	28.0	1	2	W./C. 6607	23.4500	NaN	S	-0.104579	0.000000
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C	-0.258192	-0.224833
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q	0.202648	0.449667

825 rows × 14 columns

Fig 7 :- Interquartile range

So this is the new dataframe which is cleaned according to IQR and we can see the number of row is reduced from 892 to 825 and this dataset will be used in next of the question to solve them.

Question 3:-

Based on the IQR method – remove all the observations that contain outlier for Age variable and create a new dataset: cleaned_data.

Question 3

In [18]:
```
# Create cleaned_data by removing outliers
cleaned_data = df[(df['Age'] >= lower_bound) & (df['Age'] <= upper_bound)]
cleaned_data
```

Out[18]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	zscore	modified_zscore
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	-0.565419	-0.674500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	0.663488	1.124167
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	-0.258192	-0.224833
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	0.433068	0.786917
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	0.433068	0.786917
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S	-0.181385	-0.112417
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S	-0.795839	-1.011750
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	28.0	1	2	W./C. 6607	23.4500	NaN	S	-0.104579	0.000000
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C	-0.258192	-0.224833
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q	0.202648	0.449667

825 rows × 14 columns

Fig 8 :- Cleaned - data

Question 4:-

A box plot, also known as a box-and-whisker plot, is a data visualization technique used to represent the distribution of a set of continuous or numerical data. The box plot displays the median, quartiles, and outliers of the data, allowing for a quick and easy assessment of the spread and skewness of the data. The box is drawn from the lower quartile to the upper quartile, and the median is represented as a line inside the box. Whiskers are drawn from either end of the box to the minimum and maximum data points, excluding outliers. Outliers are plotted as individual points outside the whiskers. Box plots are commonly used in exploratory data analysis and hypothesis testing.

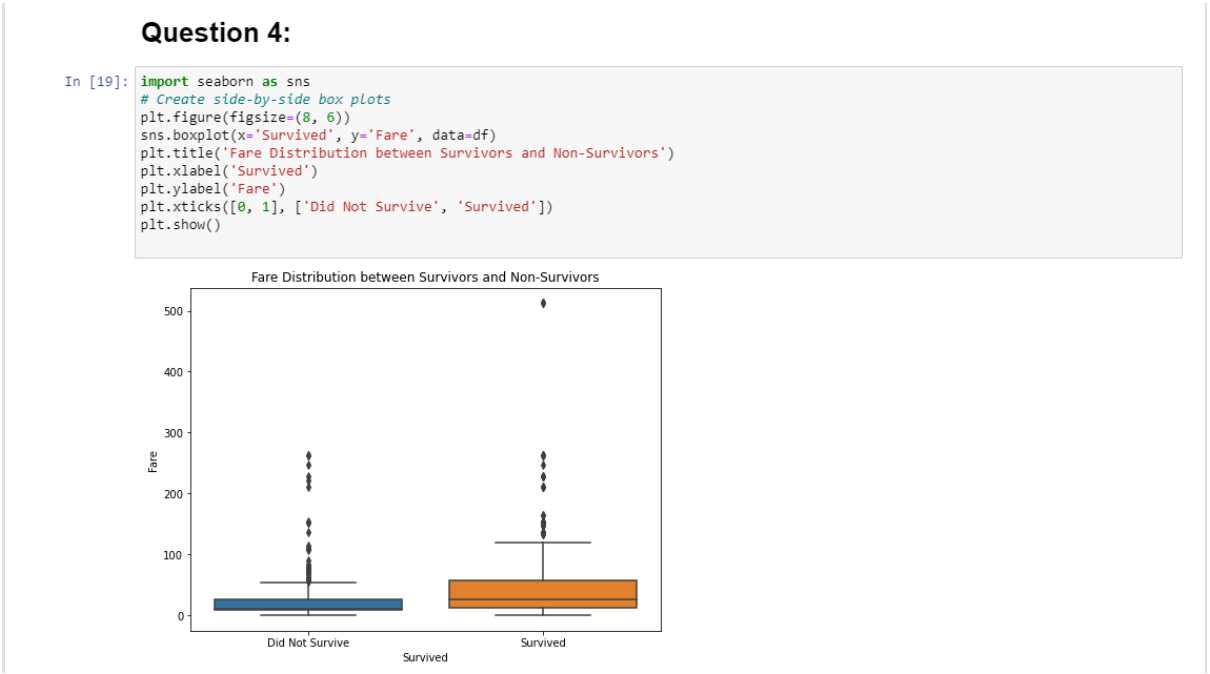


Fig 9 :- Box plot

Five number summery:
Minimum :- So the minimum value in both the graphs are nearly the same.

Q1 :- The values of q1 are also same in both the graphs

Median/Q2 :- The value of Q2 in didn't survive is nearly overlapping with its Q1 but the Q2 of the survived is distant and is defiantly ore then Q2 of the other graph.

Q3 :- Q3 of didn't survive is less then survived.

Maximum :- The max value of survived is more than the value of max in didn't survived.

Question 5 :-

a. Compute the correlation coefficient between the "Age" and "Fare" variables in the titanic dataset.

Question 5 (a):-

```
In [20]: # Compute correlation coefficient
         # Assuming 'cleaned_data' contains the cleaned dataset without outliers
         age_mean = cleaned_data['Age'].mean()
         fare_mean = cleaned_data['Fare'].mean()
         age_std = cleaned_data['Age'].std()
         fare_std = cleaned_data['Fare'].std()
         cov_age_fare = ((cleaned_data['Age'] - age_mean) * (cleaned_data['Fare'] - fare_mean)).mean()
         correlation_coefficient = cov_age_fare / (age_std * fare_std)
         correlation_coefficient

Out[20]: 0.0979026675061997
```

Fig 10 :- Correlation Coefficient

b. Use a scatter plot to visually inspect the relationship between the two variables: Age and Fare. Interpret the strength and direction of the relationship based on the computed correlation coefficient.

Question 5 (b)

```
In [21]: plt.scatter(cleaned_data['Age'], cleaned_data['Fare'])
         plt.title('Scatter Plot of Age vs. Fare')
         plt.xlabel('Age')
         plt.ylabel('Fare')
         plt.show()
```

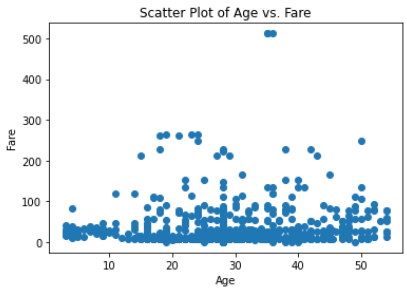


Fig 11 :- Scatter plot

The scatter plot visually shows how "Age" and "Fare" are related. You can interpret the direction (positive or negative) and the strength of the relationship based on the correlation coefficient. Positive values indicate a positive correlation, negative values indicate a negative correlation, and the magnitude indicates the strength.
So, the correlation is nearly 0.097 so it is positive but the correlation is not so strong as it near to the 0 and the graph also tells the same.

Question 6 :-

Question 6

```
In [22]: #Calculating the the mean and std of Age in cleaned data without direct formula
         age_sum = 0
         n = 0
         for i in cleaned_data["Age"]:
             n = n+1
             age_sum = age_sum + i
         age_mean = age_sum/n
         std_sum = 0

         age_variance = 0
         for age in cleaned_data['Age']:
             age_variance += (age - age_mean) ** 2
         age_std = (age_variance / n) ** 0.5
         age_std

Out[22]: 10.171085613286342

In [23]: cleaned_data["Age"].std()

Out[23]: 10.177255517167863
```

Fig 12 :- Age mean and standard deviation

```
In [24]: #Calculating the the mean and std of Age in cleaned data without direct formula
         fare_sum = 0
         n = 0
         for i in cleaned_data["Fare"]:
             n = n+1
             fare_sum = fare_sum + i
         fare_mean = fare_sum/n
         std_sum = 0

         fare_variance = 0
         for fare in cleaned_data['Fare']:
             fare_variance += (fare - fare_mean) ** 2
         fare_std = (fare_variance /n ) ** 0.5
         fare_std

Out[24]: 49.926143408502135

In [25]: cleaned_data["Fare"].std()

Out[25]: 49.95642921256646
```

Fig 13 :- Fare mean and standard deviation

```
In [26]: # Calculate the percentage of values within one standard deviation
         age_within_one_std = ((cleaned_data['Age'] >= (age_mean - age_std)) & (cleaned_data['Age'] <= (age_mean + age_std))).mean() * 10
         age_within_one_std

Out[26]: 69.6969696969697

In [27]: fare_within_one_std = ((cleaned_data['Fare'] >= (fare_mean - fare_std)) & (cleaned_data['Fare'] <= (fare_mean + fare_std))).mean(
         fare_within_one_std

Out[27]: 91.87878787878788
```

Fig 14 :- Percentage of data between standard deviation

References :-

https://saturncloud.io/blog/how-to-use-python-pandas-to-get-unique-values-ignoring-nan/

https://www.geeksforgeeks.org/bar-plot-in-matplotlib/

https://www.w3schools.com/python/matplotlib_histograms.asp

https://www.w3schools.com/python/matplotlib_line.asp

https://pythonbasics.org/seaborn-distplot/

https://seaborn.pydata.org/generated/seaborn.distplot.html

https://vitalflux.com/pandas-impute-missing-values-mean-median-mode/

https://www.geeksforgeeks.org/find-median-of-list-in-python/

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.boxplot.html