Problem statement

The number of books sold by a bookseller per day is given in 'bookseller.csv'. Let X = Number of books sold by a bookseller per day X is a Discrete Random variable (because it represents the book count). Let's see the distribution of X and answer the below questions.

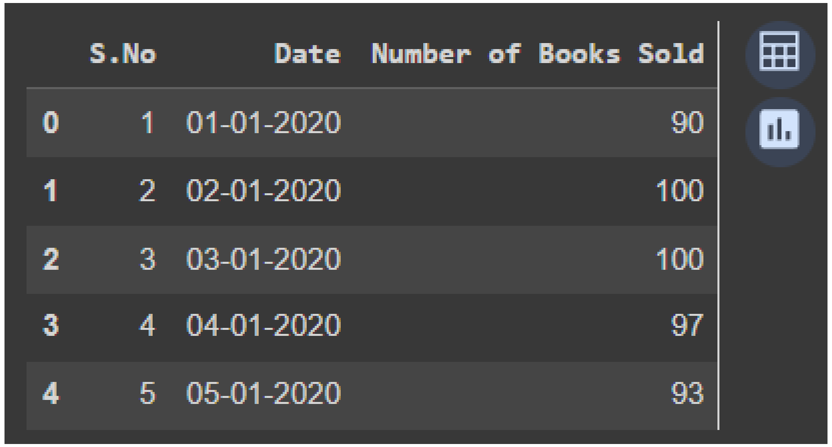
1. Find the probability that more than (or equal to) 96 books will be sold on a given day
2. Find the probability that less than (or equal to) 92 books will be sold on a given day

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv('/content/bookseller (2).csv')
```

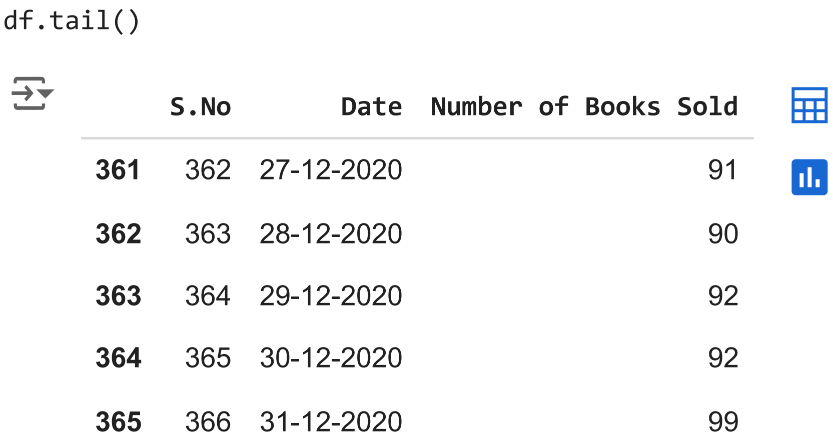
```
df.head()
```

	S.No	Date	Number of Books Sold
0	1	01-01-2020	90
1	2	02-01-2020	100
2	3	03-01-2020	100
3	4	04-01-2020	97
4	5	05-01-2020	93

Next steps: Generate code with df | View recommended plots | New interactive sheet

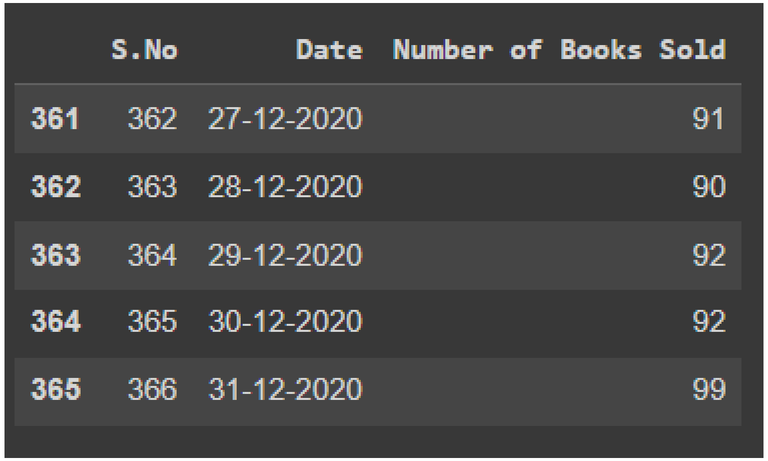


	S.No	Date	Number of Books Sold
0	1	01-01-2020	90
1	2	02-01-2020	100
2	3	03-01-2020	100
3	4	04-01-2020	97
4	5	05-01-2020	93



```
df.tail()
```

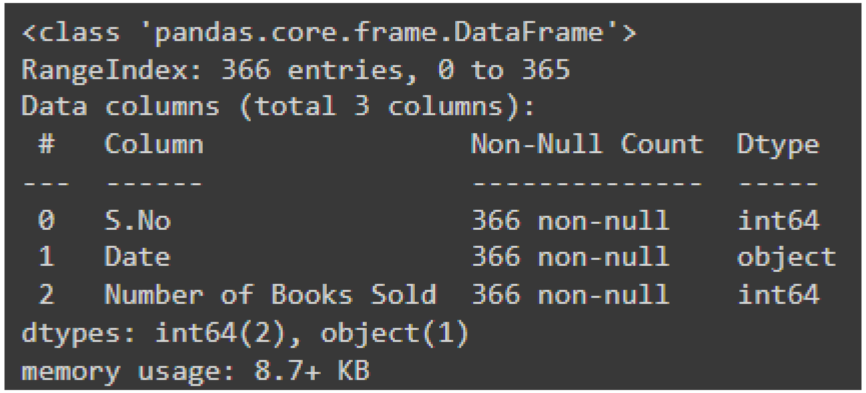
	S.No	Date	Number of Books Sold
361	362	27-12-2020	91
362	363	28-12-2020	90
363	364	29-12-2020	92
364	365	30-12-2020	92
365	366	31-12-2020	99



	S.No	Date	Number of Books Sold
361	362	27-12-2020	91
362	363	28-12-2020	90
363	364	29-12-2020	92
364	365	30-12-2020	92
365	366	31-12-2020	99

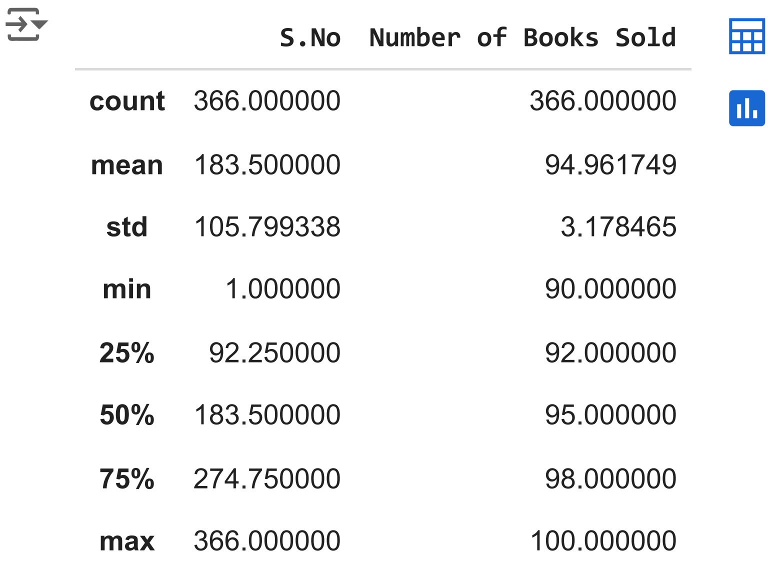
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 366 entries, 0 to 365
Data columns (total 3 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   S.No                  366 non-null    int64 
 1   Date                  366 non-null    object
 2   Number of Books Sold  366 non-null    int64 
dtypes: int64(2), object(1)
memory usage: 8.7+ KB
```

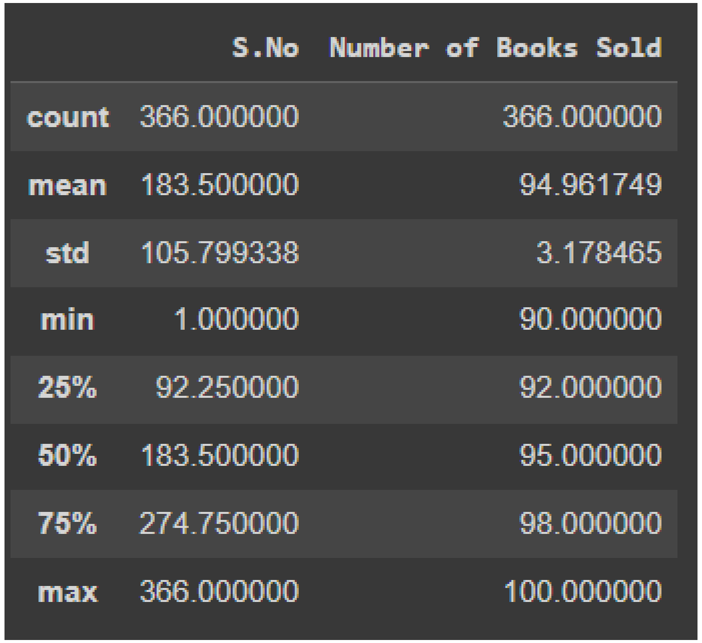


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 366 entries, 0 to 365
Data columns (total 3 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   S.No                  366 non-null    int64 
 1   Date                  366 non-null    object
 2   Number of Books Sold  366 non-null    int64 
dtypes: int64(2), object(1)
memory usage: 8.7+ KB
```

```
df.describe()
```



	S.No	Number of Books Sold
count	366.000000	366.000000
mean	183.500000	94.961749
std	105.799338	3.178465
min	1.000000	90.000000
25%	92.250000	92.000000
50%	183.500000	95.000000
75%	274.750000	98.000000
max	366.000000	100.000000

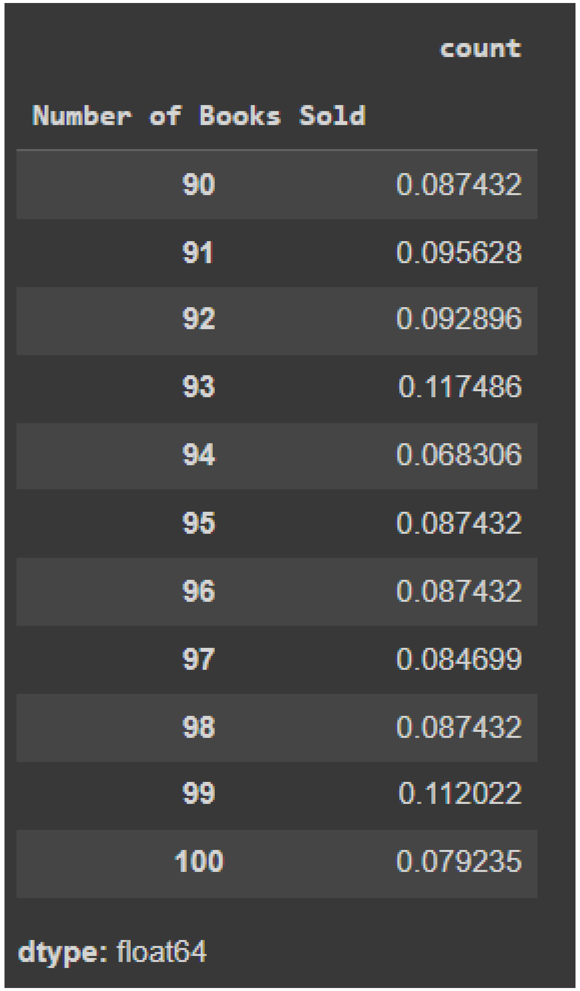


	S.No	Number of Books Sold
count	366.000000	366.000000
mean	183.500000	94.961749
std	105.799338	3.178465
min	1.000000	90.000000
25%	92.250000	92.000000
50%	183.500000	95.000000
75%	274.750000	98.000000
max	366.000000	100.000000

```
book_distribution = df['Number of Books Sold'].value_counts().sort_index()
prob_distribution = book_distribution / book_distribution.sum()
prob_distribution
```

```
                          count
Number of Books Sold
          90           0.087432
          91           0.095628
          92           0.092896
          93           0.117486
          94           0.068306
          95           0.087432
          96           0.087432
          97           0.084699
          98           0.087432
          99           0.112022
         100           0.079235

dtype: float64
```

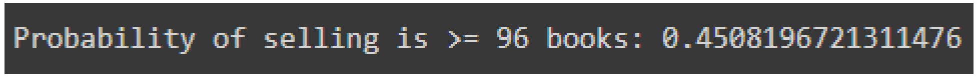


Number of Books Sold	count
90	0.087432
91	0.095628
92	0.092896
93	0.117486
94	0.068306
95	0.087432
96	0.087432
97	0.084699
98	0.087432
99	0.112022
100	0.079235

dtype: float64

```
prob_more_equal_96 = prob_distribution[prob_distribution.index >=
96].sum()
print(f"Probability of selling is >= 96 books: {prob_more_equal_96}")
```

```
Probability of selling is >= 96 books: 0.4508196721311476
```



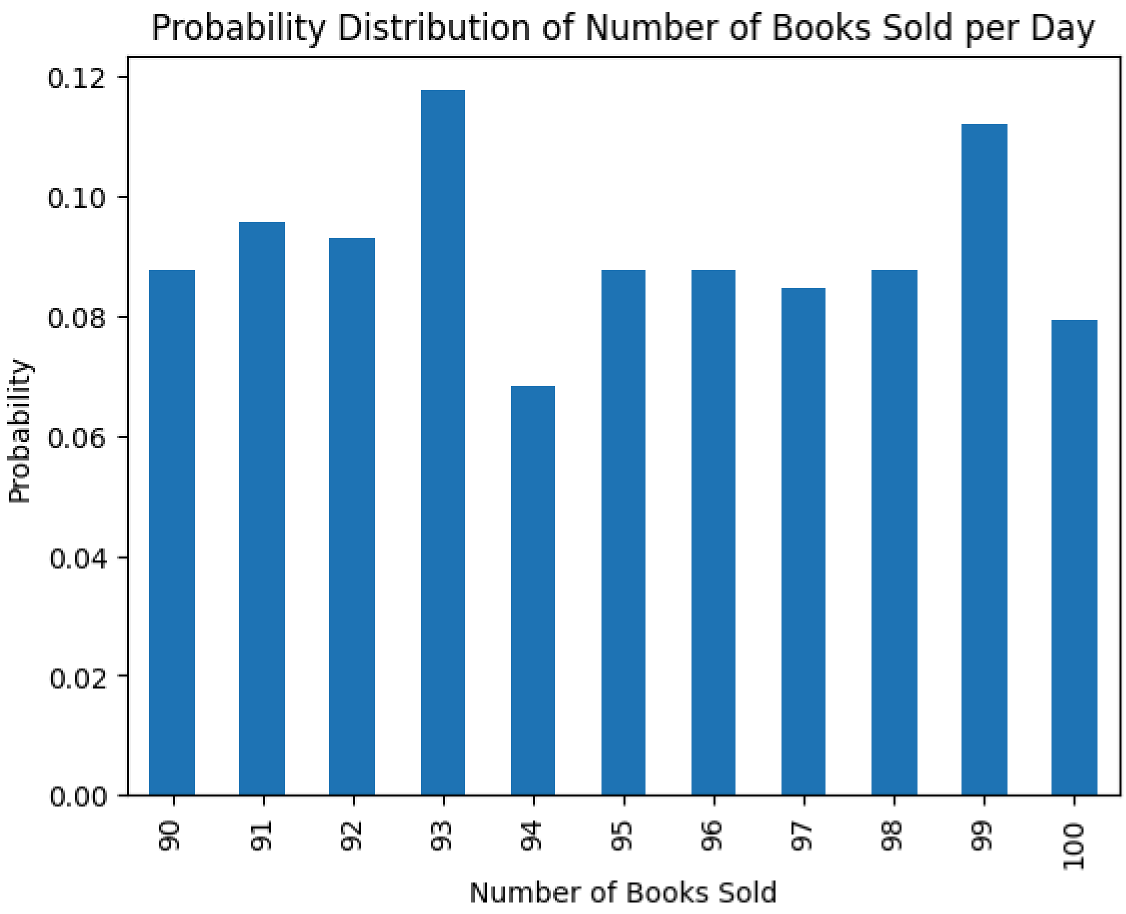
```
Probability of selling is >= 96 books: 0.4508196721311476
```

```
prob_less_equal_92 = prob_distribution[prob_distribution.index <=
92].sum()
print(f"Probability of selling <= 92 books: {prob_less_equal_92}")
```

```
Probability of selling <= 92 books: 0.27595628415300544
```

```
Probability of selling <= 92 books: 0.27595628415300544
```

```
prob_distribution.plot(kind='bar')
plt.title('Probability Distribution of Number of Books Sold per Day')
plt.xlabel('Number of Books Sold')
plt.ylabel('Probability')
plt.show()
```


Probability Distribution of Number of Books Sold per Day
Probability
0.12
0.10
0.08
0.06
0.04
0.02
0.00
90
91
92
93
94
95
96
97
98
99
100
Number of Books Sold

Problem statement

IT industry records the amount of time a software engineer needs to fix a bug in the initial phase of software development in 'debugging.csv'.

Let

X = Time needed to fix bugs

X is a continuous random variable. Let's see the distribution of X and answer the below questions.

1. Find the probability that a randomly selected software debugging requires less than three hours

2. Find the probability that a randomly selected software debugging requires more than two hours

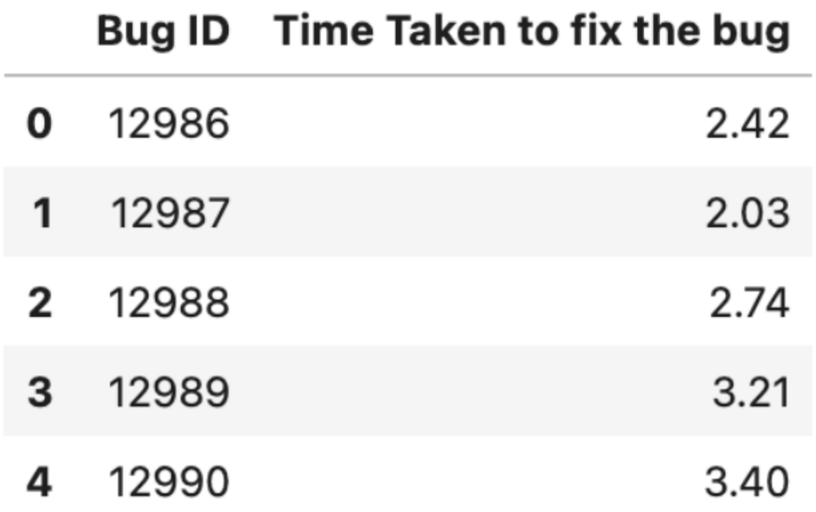
3. Find the 50th percentile of the software debugging tire

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
```

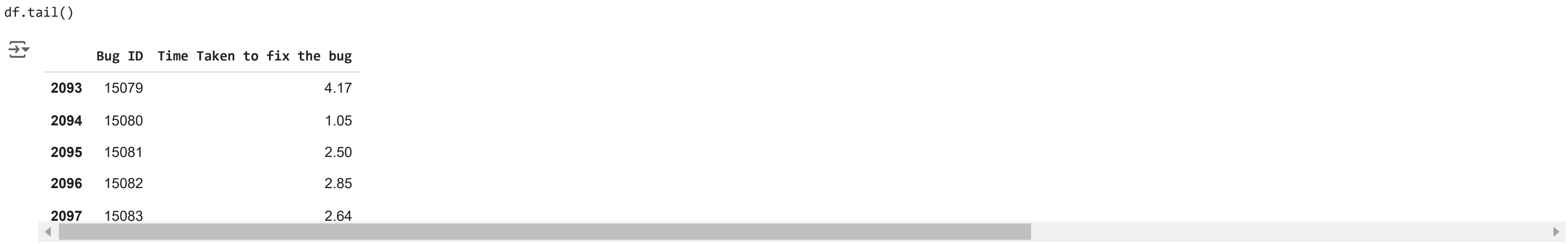
```
df = pd.read_csv('/Users/raj/Desktop/CSV files/Debugging.csv')
```

```
df.head()
```

	Bug ID	Time Taken to fix the bug
0	12986	2.42
1	12987	2.03
2	12988	2.74
3	12989	3.21
4	12990	3.40

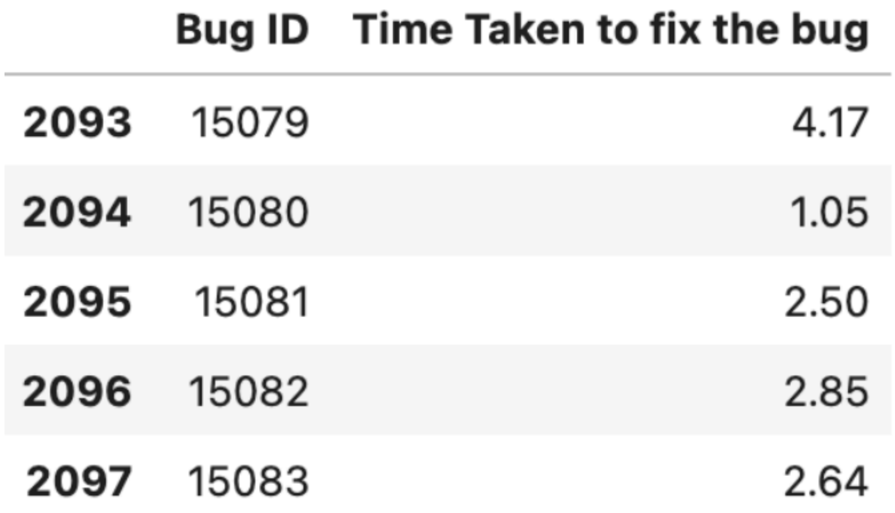


	Bug ID	Time Taken to fix the bug
0	12986	2.42
1	12987	2.03
2	12988	2.74
3	12989	3.21
4	12990	3.40

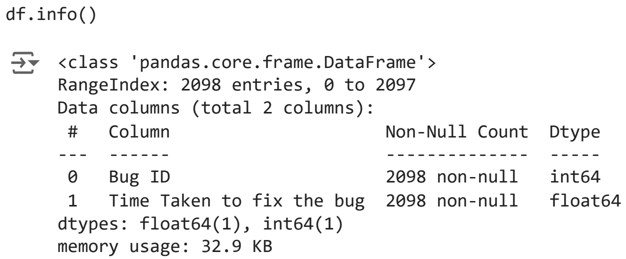


```
df.tail()
```

	Bug ID	Time Taken to fix the bug
2093	15079	4.17
2094	15080	1.05
2095	15081	2.50
2096	15082	2.85
2097	15083	2.64

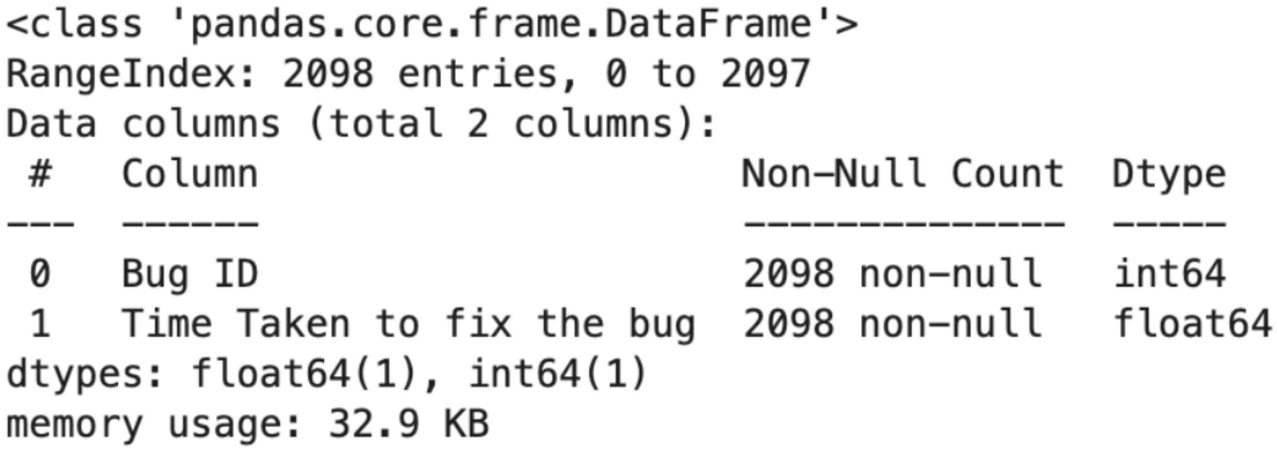


	Bug ID	Time Taken to fix the bug
2093	15079	4.17
2094	15080	1.05
2095	15081	2.50
2096	15082	2.85
2097	15083	2.64



```
df.info()
```

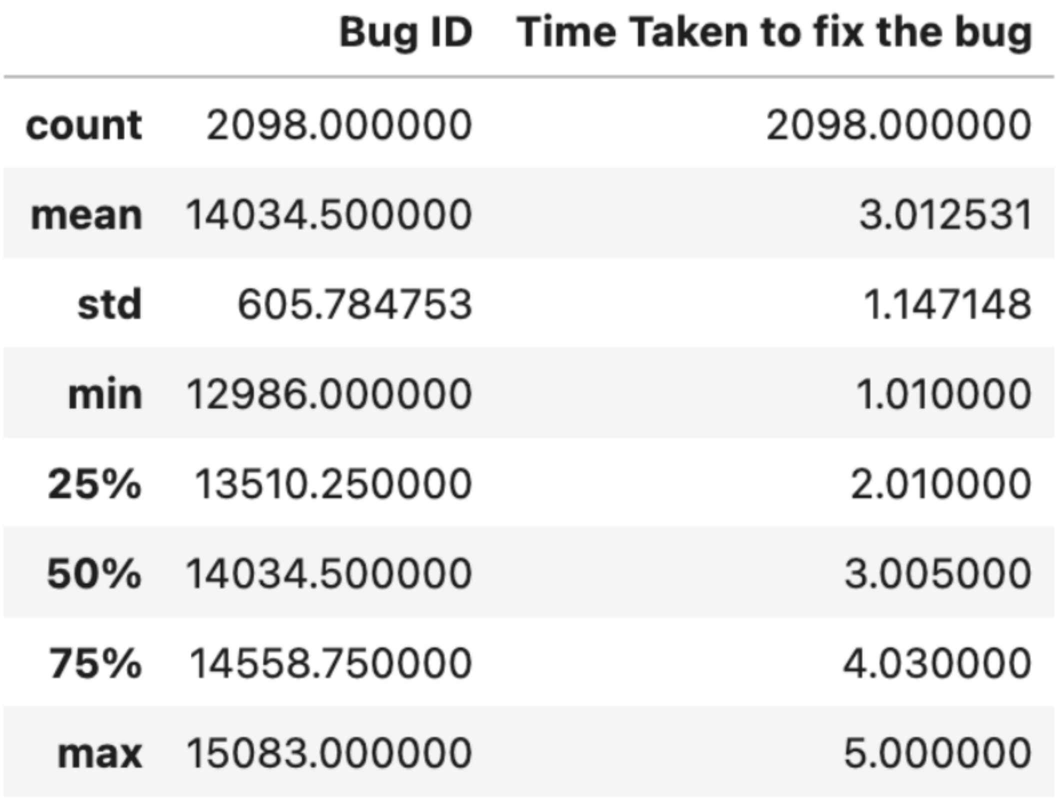
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2098 entries, 0 to 2097
Data columns (total 2 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Bug ID                     2098 non-null   int64
 1   Time Taken to fix the bug  2098 non-null   float64
dtypes: float64(1), int64(1)
memory usage: 32.9 KB
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2098 entries, 0 to 2097
Data columns (total 2 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Bug ID                     2098 non-null   int64
 1   Time Taken to fix the bug  2098 non-null   float64
dtypes: float64(1), int64(1)
memory usage: 32.9 KB
```

```
df.describe()
```

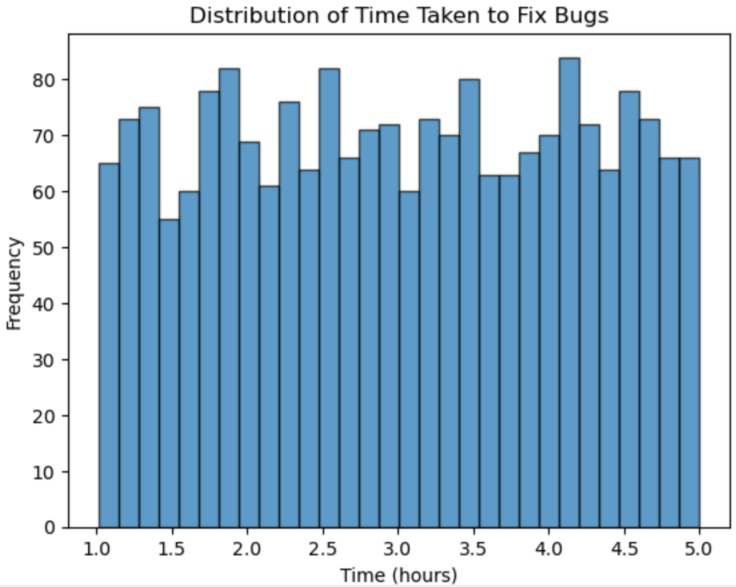
```
             Bug ID  Time Taken to fix the bug
count   2098.000000                2098.000000
mean   14034.500000                   3.012531
std      605.784753                   1.147148
min    12986.000000                   1.010000
25%    13510.250000                   2.010000
50%    14034.500000                   3.005000
75%    14558.750000                   4.030000
max    15083.000000                   5.000000
```

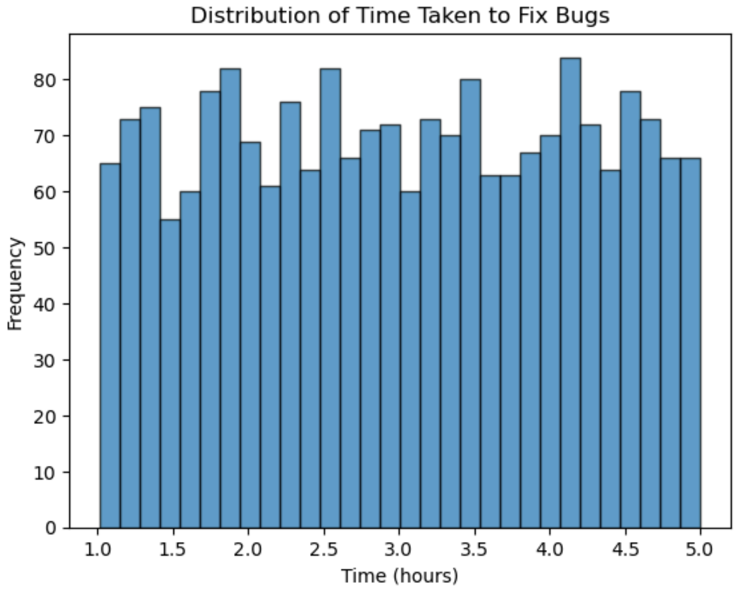


	Bug ID	Time Taken to fix the bug
count	2098.000000	2098.000000
mean	14034.500000	3.012531
std	605.784753	1.147148
min	12986.000000	1.010000
25%	13510.250000	2.010000
50%	14034.500000	3.005000
75%	14558.750000	4.030000
max	15083.000000	5.000000

Check the Distribution of 'Time Taken to Fix the Bug'

```
# Plot a histogram to visualize the distribution
plt.hist(df['Time Taken to fix the bug'], bins=30, edgecolor='k', alpha=0.7)
plt.title('Distribution of Time Taken to Fix Bugs')
plt.xlabel('Time (hours)')
plt.ylabel('Frequency')
plt.show()
```

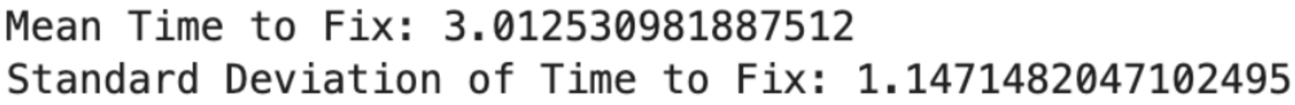




Calculate the Mean and Standard Deviation

```
mean_time = df['Time Taken to fix the bug'].mean()
std_time = df['Time Taken to fix the bug'].std()
print(f"Mean Time to Fix: {mean_time}")
print(f"Standard Deviation of Time to Fix: {std_time}")
```

```
Mean Time to Fix: 3.012530981887512
Standard Deviation of Time to Fix: 1.1471482047102495
```

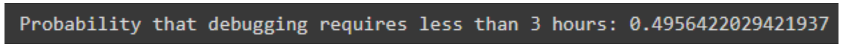


```
Mean Time to Fix: 3.012530981887512
Standard Deviation of Time to Fix: 1.1471482047102495
```

Find the probability that Time Taken to Fix is less than 3 hours

```
prob_less_than_3 = stats.norm.cdf(3, loc=mean_time, scale=std_time)
print(f"Probability that debugging requires less than 3 hours: {prob_less_than_3}")
```

```
Probability that debugging requires less than 3 hours: 0.4956422029421937
```



```
Probability that debugging requires less than 3 hours: 0.4956422029421937
```