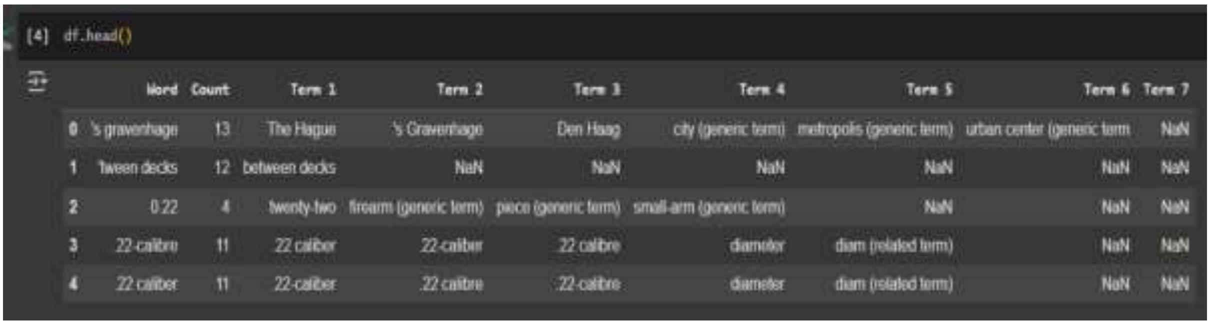
Name:-Harshal Patil
PRN:-202401070048
Roll no:-ET1-37
Dataset:-WordNet

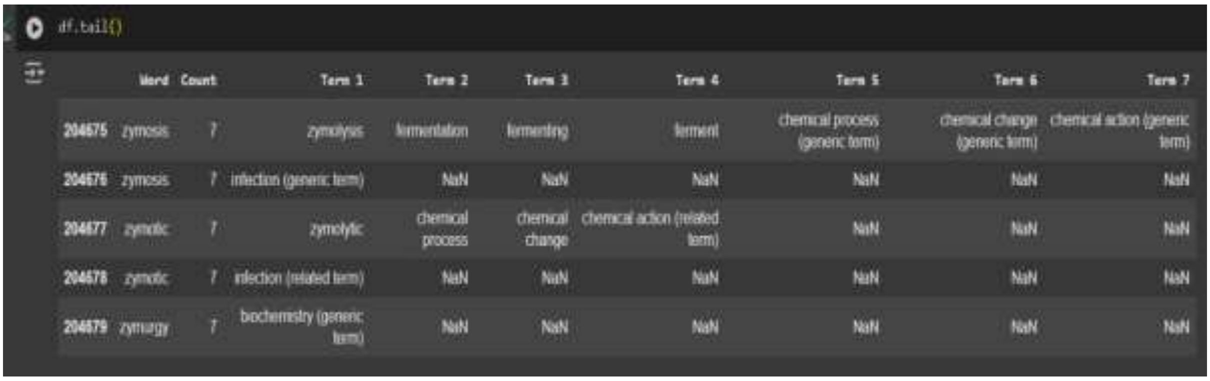
1. Display 1st five rows of the data set



```
[4] df.head()
```

	Word	Count	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6	Term 7
0	's gravenhage	13	The Hague	's Gravenhage	Den Haag	city (generic term)	metropolis (generic term)	urban center (generic term	NaN
1	'tween decks	12	between decks	NaN	NaN	NaN	NaN	NaN	NaN
2	0.22	4	twenty-two	firearm (generic term)	piece (generic term)	small-arm (generic term)	NaN	NaN	NaN
3	22-calibre	11	22 caliber	22-caliber	22 calibre	diameter	diam (related term)	NaN	NaN
4	22 caliber	11	22-caliber	.22 calibre	.22-calibre	diameter	diam (related term)	NaN	NaN

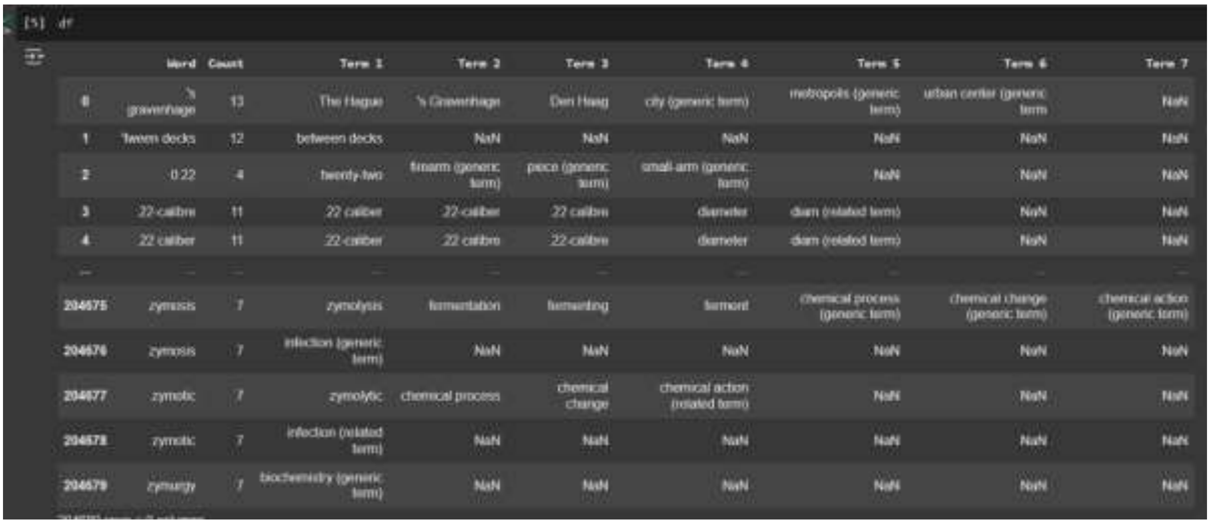
2. Display last 5 rows of the data set



```
df.tail()
```

	Word	Count	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6	Term 7
204675	zymosis	7	zymolysis	fermentation	fermenting	ferment	chemical process (generic term)	chemical change (generic term)	chemical action (generic term)
204676	zymosis	7	infection (generic term)	NaN	NaN	NaN	NaN	NaN	NaN
204677	zymotic	7	zymolytic	chemical process	chemical change	chemical action (related term)	NaN	NaN	NaN
204678	zymotic	7	infection (related term)	NaN	NaN	NaN	NaN	NaN	NaN
204679	zymurgy	7	biochemistry (generic term)	NaN	NaN	NaN	NaN	NaN	NaN

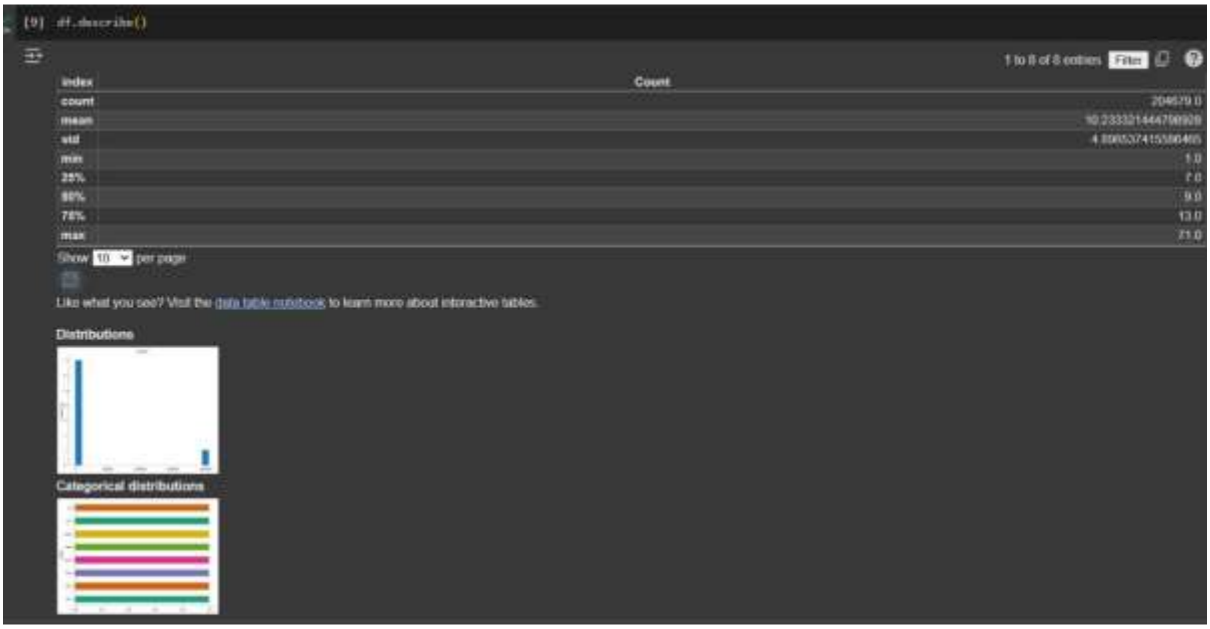
3. Easy to analyze, filter, and visualize large datasets.



```
[5] df
```

	Word	Count	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6	Term 7
0	's gravenhage	13	The Hague	's Gravenhage	Den Haag	city (generic term)	metropolis (generic term)	urban center (generic term	NaN
1	'tween decks	12	between decks	NaN	NaN	NaN	NaN	NaN	NaN
2	0.22	4	twenty-two	firearm (generic term)	piece (generic term)	small-arm (generic term)	NaN	NaN	NaN
3	22-calibre	11	22 caliber	22-caliber	22 calibre	diameter	diam (related term)	NaN	NaN
4	22 caliber	11	22-caliber	22 calibre	22-calibre	diameter	diam (related term)	NaN	NaN
...	...	...	...	...	...	...	...	...	...
204575	zymosis	7	zymolysis	fermentation	fermenting	ferment	chemical process (generic term)	chemical change (generic term)	chemical action (generic term)
204576	zymosis	7	infection (generic term)	NaN	NaN	NaN	NaN	NaN	NaN
204577	zymotic	7	zymolytic	chemical process	chemical change	chemical action (related term)	NaN	NaN	NaN
204578	zymotic	7	infection (related term)	NaN	NaN	NaN	NaN	NaN	NaN
204579	zymurgy	7	biochemistry (generic term)	NaN	NaN	NaN	NaN	NaN	NaN

4. To describe the dataset



```
[9] df.describe()
```

1 to 8 of 8 entries Filter

index	Count
count	204579.0
mean	10.23332144479892
std	4.898537415586465
min	1.0
25%	7.0
50%	9.0
75%	13.0
max	71.0

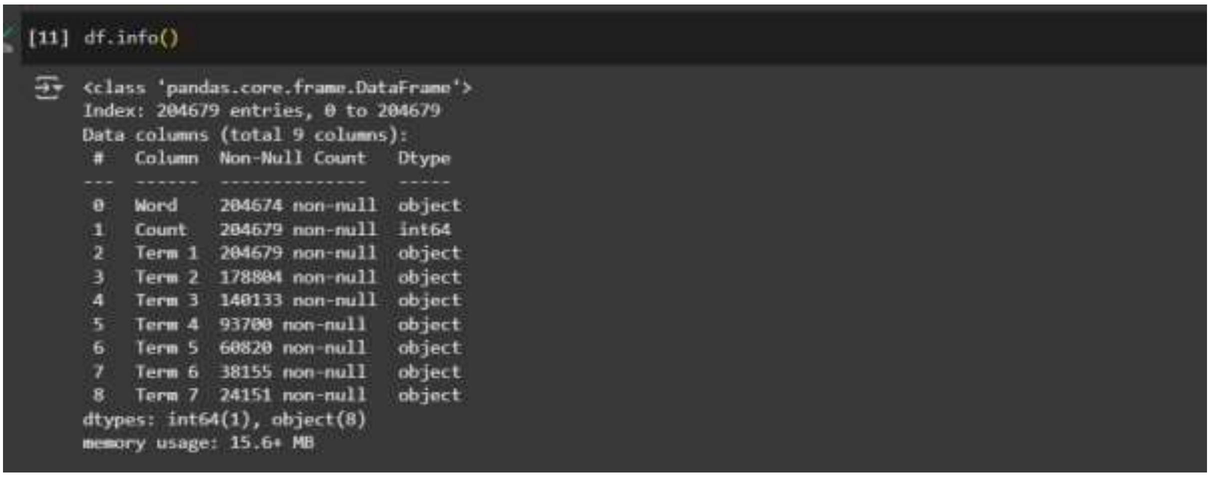
Show 10 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Distributions

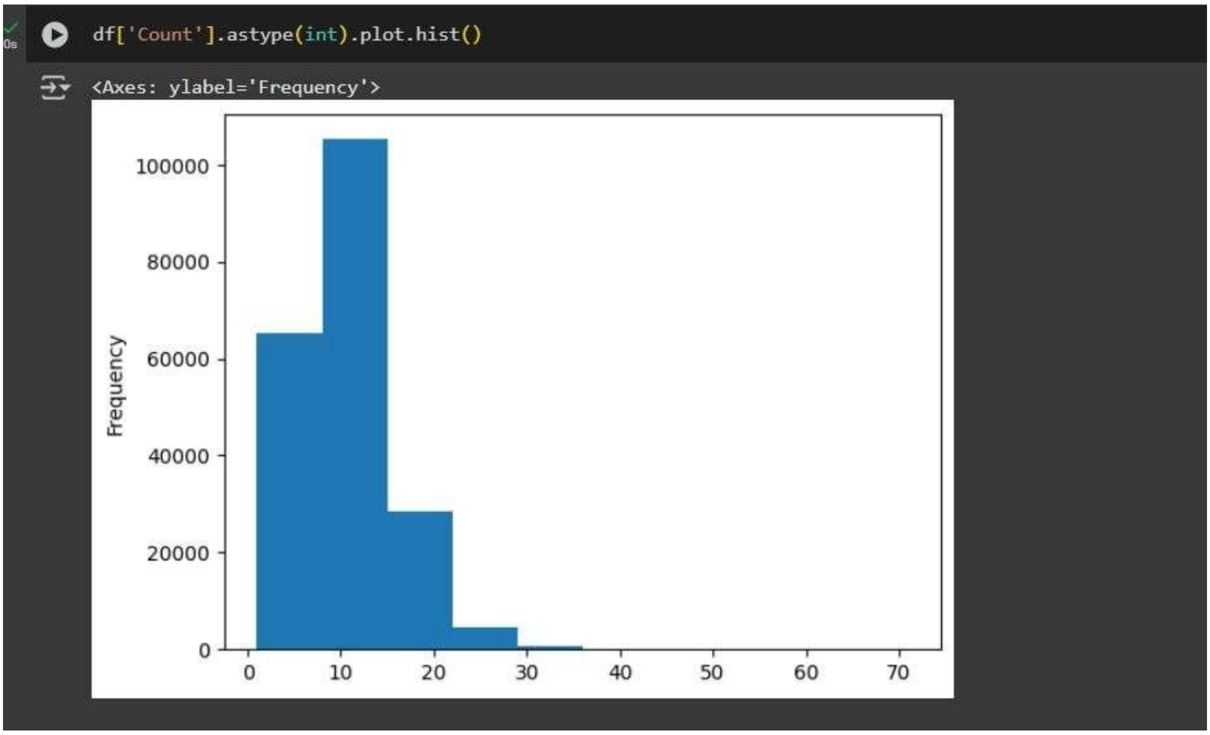
Categorical distributions

5. To display information about dataset



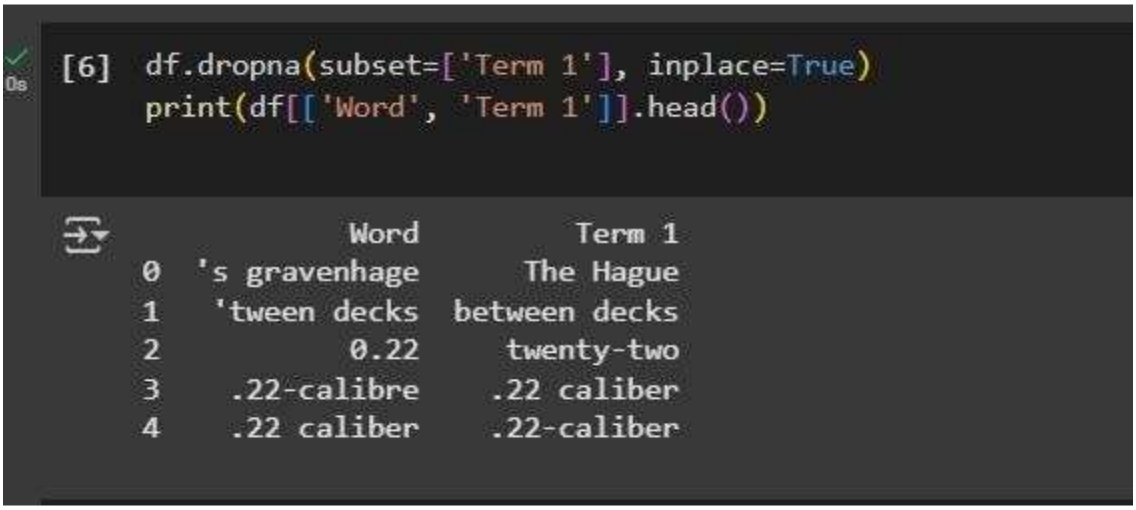
```
[11] df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 204679 entries, 0 to 204679
Data columns (total 9 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Word    204674 non-null  object
 1   Count   204679 non-null  int64
 2   Term 1  204679 non-null  object
 3   Term 2  178804 non-null  object
 4   Term 3  140133 non-null  object
 5   Term 4  93700 non-null   object
 6   Term 5  60820 non-null   object
 7   Term 6  38155 non-null   object
 8   Term 7  24151 non-null   object
dtypes: int64(1), object(8)
memory usage: 15.6+ MB
```

6. Distribution of count values



```
df['Count'].astype(int).plot.hist()

<Axes: ylabel='Frequency'>
```

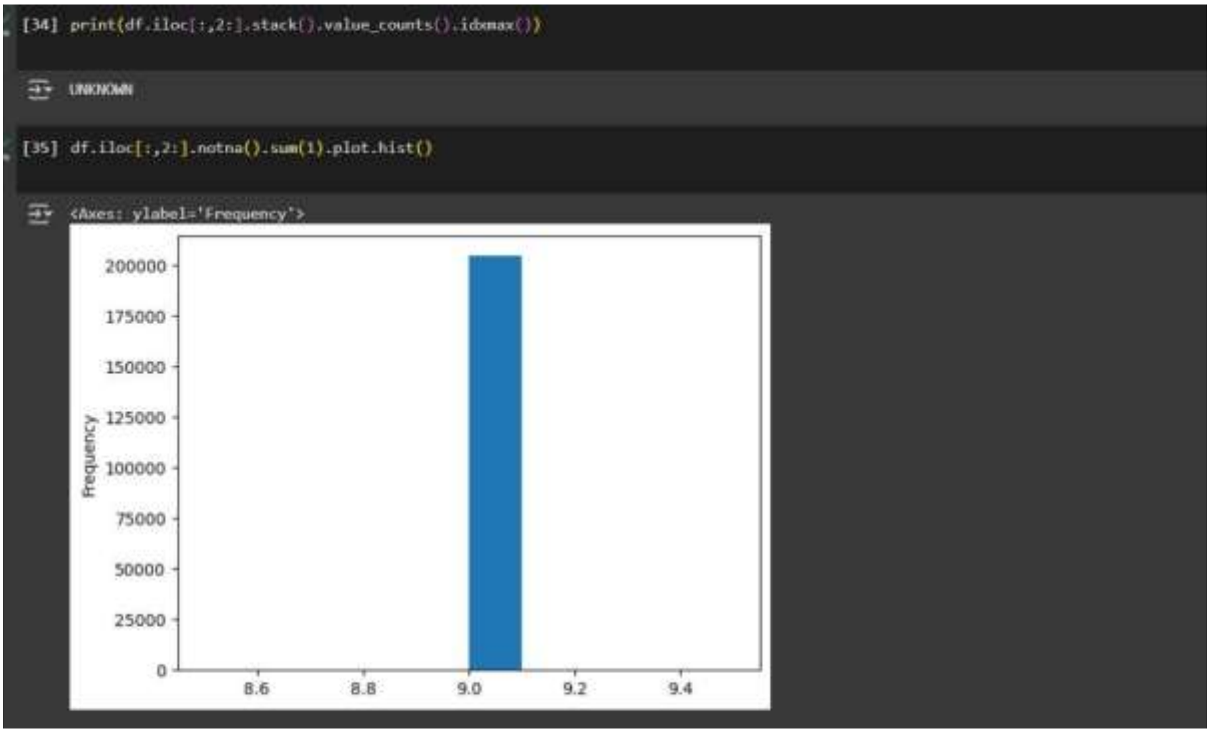
7. To clean the data by removing missing translation



```
[6] df.dropna(subset=['Term 1'], inplace=True)
    print(df[['Word', 'Term 1']].head())
```

```
              Word          Term 1
0   's gravenhage       The Hague
1     'tween decks  between decks
2             0.22      twenty-two
3      .22-calibre     .22 caliber
4      .22 caliber     .22-caliber
```

8. Find the most common value in the columns
9. To plot a histogram showing how many non missings entries each row



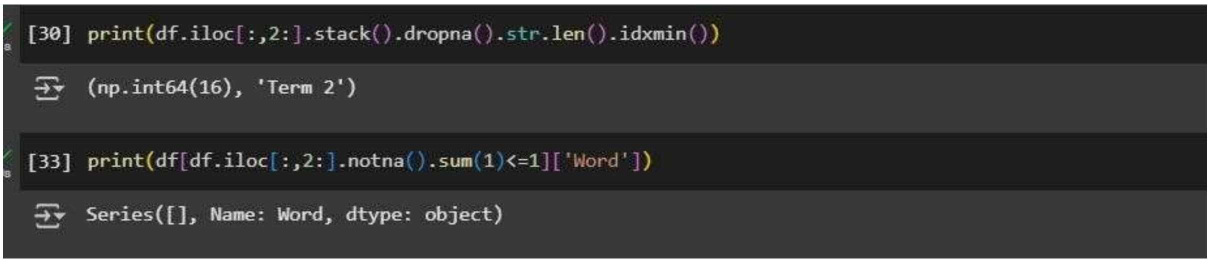
```
[34] print(df.iloc[:,2:].stack().value_counts().idxmax())
```

```
UNKNOWN
```

```
[35] df.iloc[:,2:].notna().sum(1).plot.hist()
```

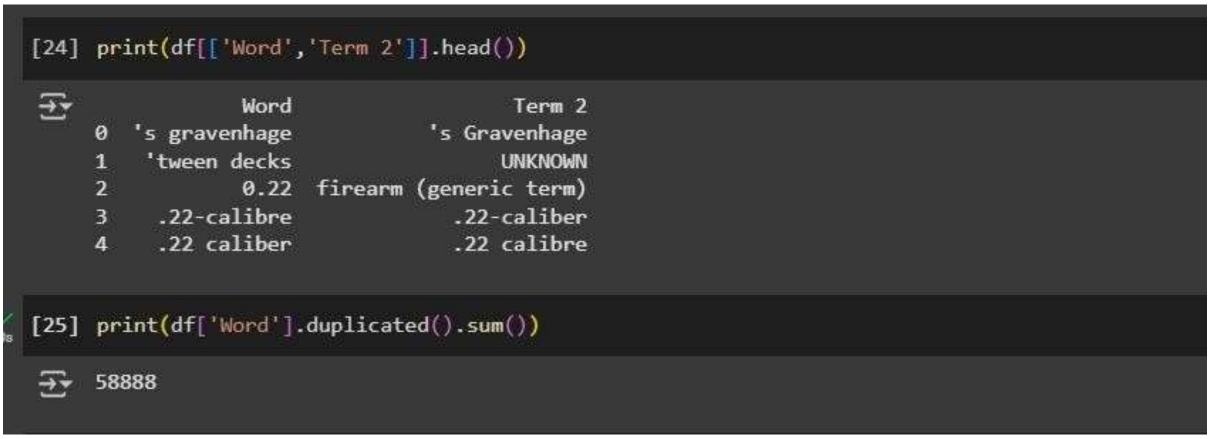
```
<Axes: ylabel='Frequency'>
```

10. To find the cell that has shortest text length
11. To find rows where there is only 0-1non null value



```
[30] print(df.iloc[:,2:].stack().dropna().str.len().idxmin())

     (np.int64(16), 'Term 2')

[33] print(df[df.iloc[:,2:].notna().sum(1)<=1]['Word'])

     Series([], Name: Word, dtype: object)
```

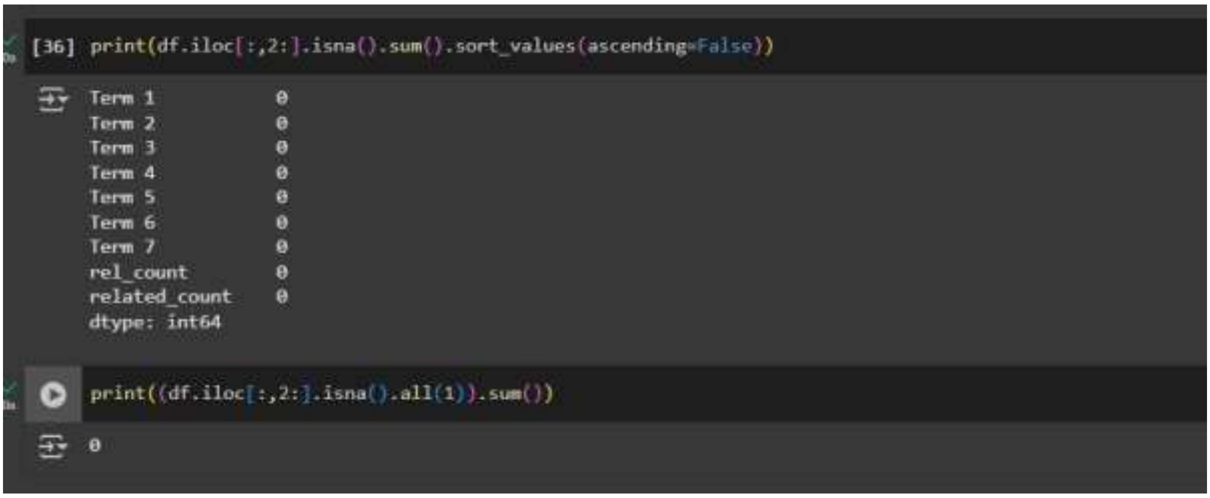
12. Display first 5 rows of the columns
13. **counts** how many times the 'Word' column has **duplicate entries.**



```
[24] print(df[['Word','Term 2']].head())

                 Word                 Term 2
     0  's gravenhage          's Gravenhage
     1   'tween decks                UNKNOWN
     2           0.22  firearm (generic term)
     3    .22-calibre            .22-caliber
     4    .22 caliber            .22 calibre

[25] print(df['Word'].duplicated().sum())

     58888
```

14. **counts the number of missing (NaN) values in each column**

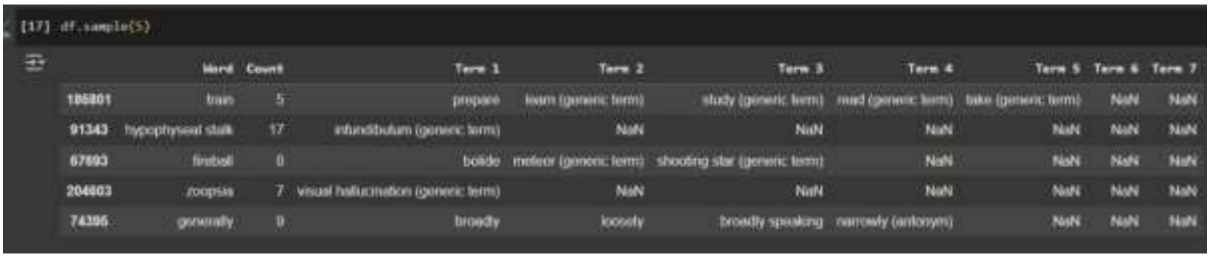
15. **counts how many rows have *all* values missing** across the selected columns



```
[36] print(df.iloc[:,2:].isna().sum().sort_values(ascending=False))

Term 1           0
Term 2           0
Term 3           0
Term 4           0
Term 5           0
Term 6           0
Term 7           0
rel_count        0
related_count    0
dtype: int64
```

```
print((df.iloc[:,2:].isna().all(1)).sum())

0
```

16. **randomly inspecting** 5 words and their related terms

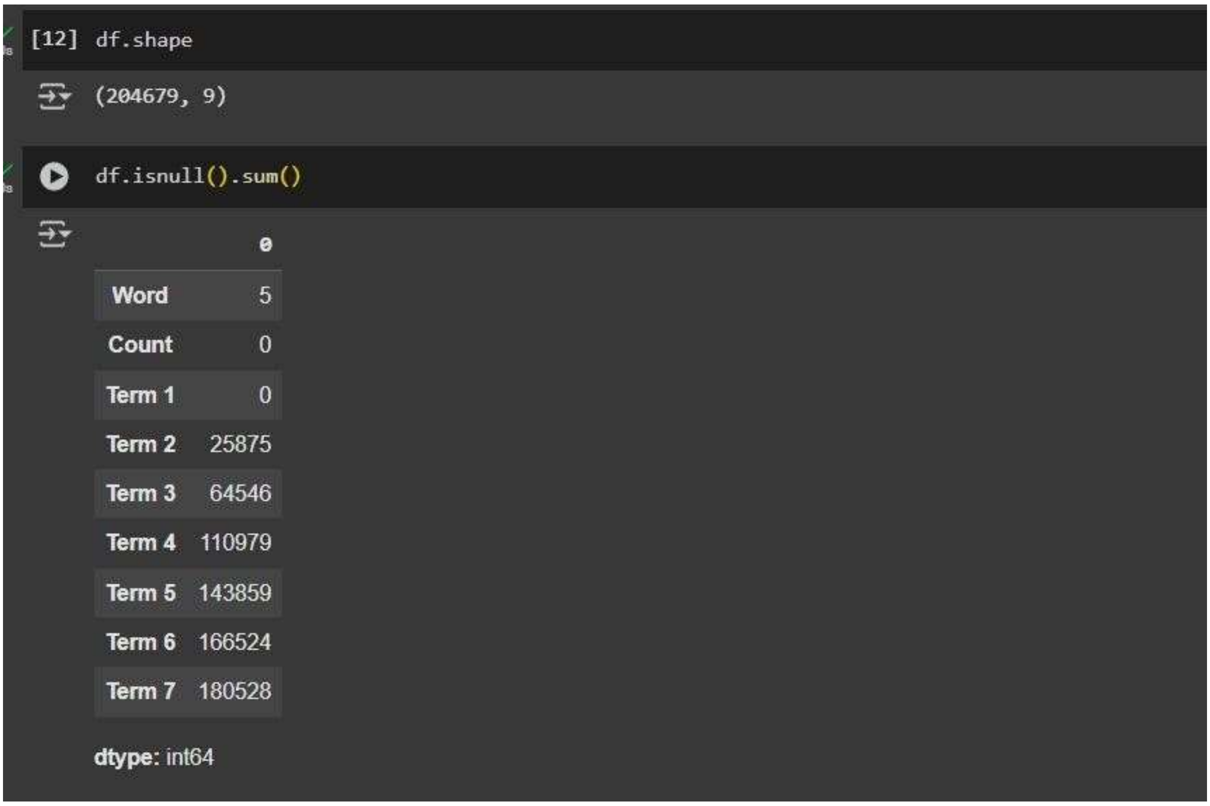


```
[17] df.sample(5)
```

	Word	Count	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6	Term 7
186801	train	5	prepare	learn (generic term)	study (generic term)	read (generic term)	take (generic term)	NaN	NaN
91343	hypophyseal stalk	17	infundibulum (generic term)	NaN	NaN	NaN	NaN	NaN	NaN
67693	fireball	0	bolide	meteor (generic term)	shooting star (generic term)	NaN	NaN	NaN	NaN
204603	zoopsia	7	visual hallucination (generic term)	NaN	NaN	NaN	NaN	NaN	NaN
74296	generally	0	broadly	loosely	broadly speaking	narrowly (antonym)	NaN	NaN	NaN

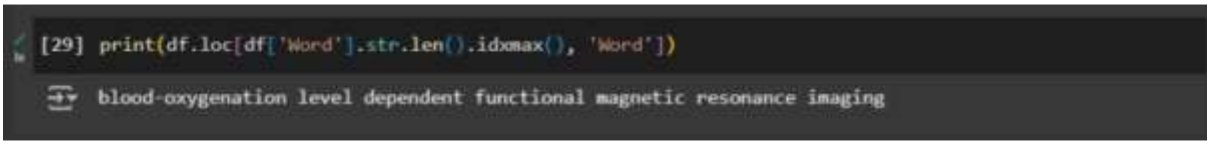
17. returns the **shape** of the DataFrame

18. checks for **missing values**



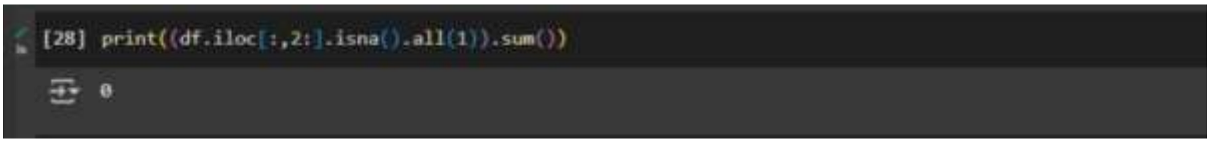
```
[12] df.shape

(204679, 9)

df.isnull().sum()

              0
Word          5
Count         0
Term 1        0
Term 2    25875
Term 3    64546
Term 4   110979
Term 5   143859
Term 6   166524
Term 7   180528

dtype: int64
```

19. calculates the **length** of each string in the **Word** column.



```
[29] print(df.loc[df['Word'].str.len().idxmax(), 'Word'])

blood-oxygenation level dependent functional magnetic resonance imaging
```

20. selects all **columns starting from the 3rd column** onward



```
[28] print((df.iloc[:,2:].isna().all(1)).sum())

0
```