

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
public=pd.read_csv("Downloads/survey.csv")
print(public)
```

	Timestamp	Age	Gender	Country	state	self_employed	\
0	2014-08-27 11:29:31	37	Female	United States	IL	NaN	
1	2014-08-27 11:29:37	44	M	United States	IN	NaN	
2	2014-08-27 11:29:44	32	Male	Canada	NaN	NaN	
3	2014-08-27 11:29:46	31	Male	United Kingdom	NaN	NaN	
4	2014-08-27 11:30:22	31	Male	United States	TX	NaN	
...	
1254	2015-09-12 11:17:21	26	male	United Kingdom	NaN	No	
1255	2015-09-26 01:07:35	32	Male	United States	IL	No	
1256	2015-11-07 12:36:58	34	male	United States	CA	No	
1257	2015-11-30 21:25:06	46	f	United States	NC	No	
1258	2016-02-01 23:04:31	25	Male	United States	IL	No	

	family_history	treatment	work_interfere	no_employees	...	\
0	No	Yes	Often	6-25	...	
1	No	No	Rarely	More than 1000	...	
2	No	No	Rarely	6-25	...	
3	Yes	Yes	Often	26-100	...	
4	No	No	Never	100-500	...	
...	
1254	No	Yes	NaN	26-100	...	
1255	Yes	Yes	Often	26-100	...	
1256	Yes	Yes	Sometimes	More than 1000	...	
1257	No	No	NaN	100-500	...	
1258	Yes	Yes	Sometimes	26-100	...	

	leave	mental_health_consequence	phys_health_consequence	\
0	Somewhat easy	No	No	
1	Don't know	Maybe	No	
2	Somewhat difficult	No	No	
3	Somewhat difficult	Yes	Yes	
4	Don't know	No	No	
...	
1254	Somewhat easy	No	No	
1255	Somewhat difficult	No	No	
1256	Somewhat difficult	Yes	Yes	
1257	Don't know	Yes	No	
1258	Don't know	Maybe	No	

	coworkers	supervisor	mental_health_interview	\
0	Some of them	Yes	No	
1	No	No	No	
2	Yes	Yes	Yes	
3	Some of them	No	Maybe	
4	Some of them	Yes	Yes	
...	
1254	Some of them	Some of them	No	
1255	Some of them	Yes	No	
1256	No	No	No	
1257	No	No	No	
1258	Some of them	No	No	

	phys_health_interview	mental_vs_physical	obs_consequence	comments
0	Maybe	Yes	No	NaN
1	No	Don't know	No	NaN
2	Yes	No	No	NaN
3	Maybe	No	Yes	NaN
4	Yes	Don't know	No	NaN
...

1254	No	Don't know	No	NaN
1255	No	Yes	No	NaN
1256	No	No	No	NaN
1257	No	No	No	NaN
1258	No	Don't know	No	NaN

```
[1259 rows x 27 columns]
std = np.std(public)
print(std)
Age      2.817180e+09
dtype: float64
C:\Users\HP\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3579:
FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError. Select only valid columns before calling the reduction.
    return std(axis=axis, dtype=dtype, out=out, ddof=ddof, **kwargs)
mean = np.mean(public)
print(mean)
Age      7.942815e+07
dtype: float64
C:\Users\HP\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3438:
FutureWarning: In a future version, DataFrame.mean(axis=None) will return a
scalar mean over the entire DataFrame. To retain the old behavior, use
'frame.mean(axis=0)' or just 'frame.mean()'
    return mean(axis=axis, dtype=dtype, out=out, **kwargs)
C:\Users\HP\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3438:
FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError. Select only valid columns before calling the reduction.
    return mean(axis=axis, dtype=dtype, out=out, **kwargs)
from matplotlib import pyplot as plt
x = mean
y = std
plt.plot(mean, std)
plt.title("Line graph")
plt.ylabel('Y axis')
plt.xlabel('X axis')
plt.show()

x_column = public["Age"]
y_column = public["Timestamp"]

# Create the scatter plot
plt.scatter(x_column, y_column,)

# Add a title and axis labels
plt.title("Scatter Plot of {} vs. {}".format(x_column.name, y_column.name ))
plt.xlabel(x_column.name)
plt.ylabel(y_column.name)

# Show the plot
plt.show()

column_1 = public["no_employees"]
column_2 = public["Age"]

# Create the histogram
fig, axs = plt.subplots(1, 2)

axs[0].hist(column_1)
axs[1].hist(column_2)
```

```
# Add a title and axis labels for each subplot
axs[0].set_title("Histogram of {} Column".format(column_1.name))
axs[1].set_title("Histogram of {} Column".format(column_2.name))

axs[0].set_xlabel(column_1.name)
axs[1].set_xlabel(column_2.name)

axs[0].set_ylabel("Count")
axs[1].set_ylabel("Count")

# Adjust the subplot layout
plt.tight_layout()

# Show the plot
plt.show()
```