```
import pandas as pd
import seaborn as sns


dt = pd.read_csv('/content/iris.csv')


dt
```

Release notes NameErrc •••

Please explain the error:

```
NameError: name 'dt' is not d
```

Hypothesis

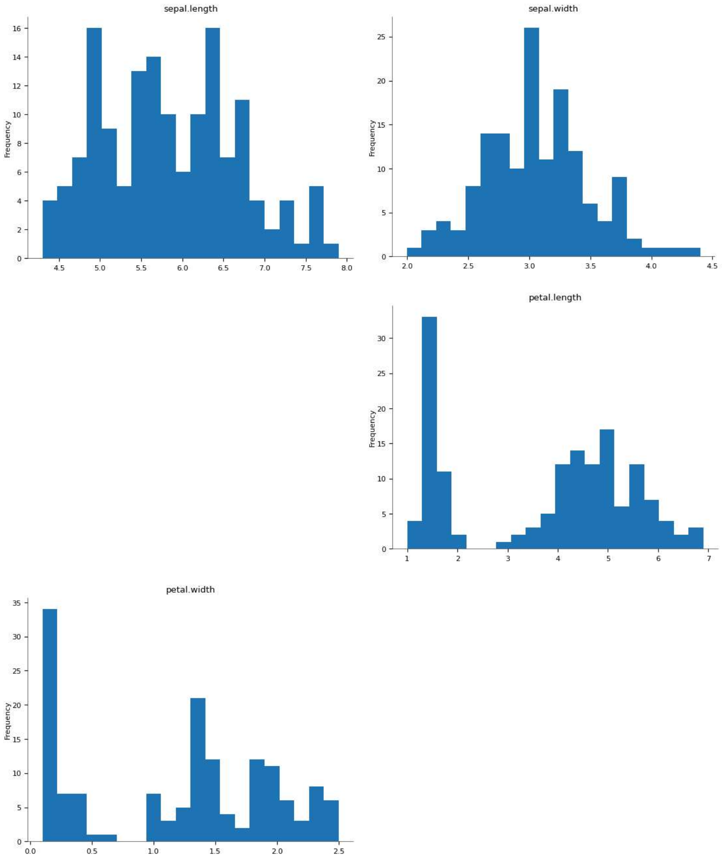
The `NameError: name 'dt' is not defined` indicates that the variable `dt` was not found in the current scope when the code `dt` was executed. This is likely because the cell where `dt = pd.read_csv('/content/iris.csv` was defined was not run, or its output was cleared from the environment.

Based on the provided source files and traceback, it appears that the cells were executed in the following order:

1. `import pandas as pd\nimport seaborn as sns` (Cell with source file `ipython-input-3-36673dcb6ee0`) - This imports the necessary libraries.
2. `dt` (Cell with source file `ipython-input-5-36673dcb6ee0`) - This cell attempts to display the `dt` variable. However, `dt` has not been defined yet.
3. `dt = pd.read_csv('/content/iri` (Cell with source file `ipython-input-0-36673dcb6ee0`) - This cell,

150 rows × 5 columns

Distributions



Categorical distributions

Virginica
Versicolor
variety

which *defines* `dt`, was executed *after* the cell that tried to display it.

Therefore, when the cell with `dt` was executed, the variable `dt` did not exist in the global namespace, leading to the `NameError`.

Suggested Changes

To fix this issue, ensure that the cell defining `dt` is executed before the cell that attempts to use it. The corrected order of execution should be:

1. Import libraries.
2. Load the CSV file into `dt`.
3. Display `dt`.

Since you are working in a Jupyter notebook, simply re-running the cells in the correct order should resolve the problem. Below is the code in the correct execution order, assuming each block is a separate cell in your notebook.

```
# Import necessary libraries
import pandas as pd
import seaborn as sns
```

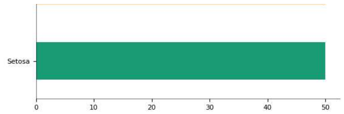
Use code with caution

```
# Load the dataset into the v
dt = pd.read_csv('/content/ir
```

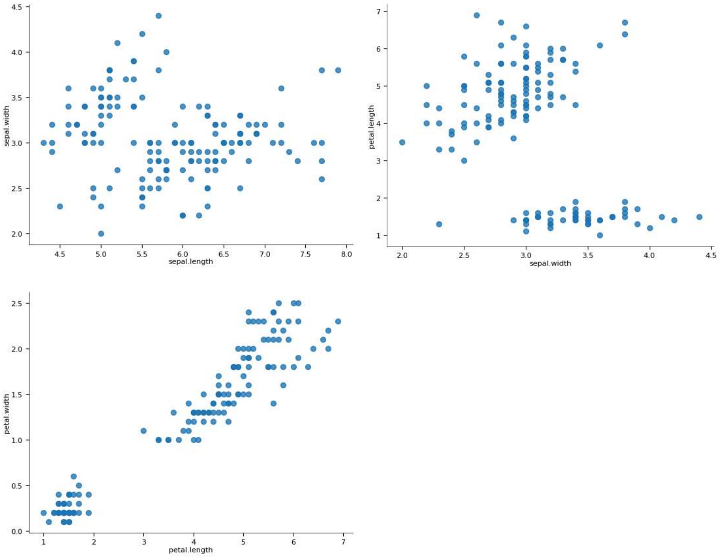
Use code with caution

```
# Display the content of dt
dt
```

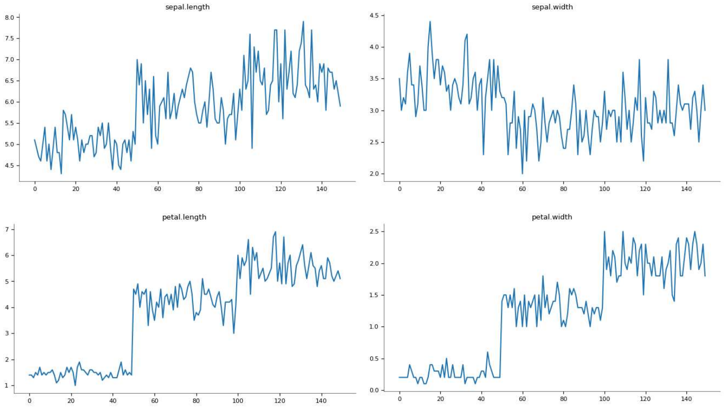
Use code with caution



2-d distributions



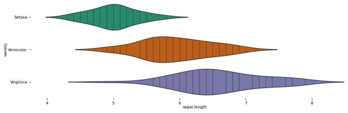
Values



Faceted distributions

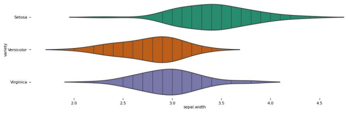
```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is depre
```



```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is depre
```



```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is depre
```

Rate this answer

Next steps: Generate code with dt View recommended plots

```
dt.describe()
```

	sepal.length	sepal.width	petal.length	peta
count	150.000000	150.000000	150.000000	15
mean	5.843333	3.057333	3.758000	
std	0.828066	0.435866	1.765298	
min	4.300000	2.000000	1.000000	
25%	5.100000	2.800000	1.600000	
50%	5.800000	3.000000	4.350000	
75%	6.400000	3.300000	5.100000	
max	7.900000	4.400000	6.900000	

```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal.length  150 non-null    float64
 1   sepal.width   150 non-null    float64
 2   petal.length  150 non-null    float64
 3   petal.width   150 non-null    float64
 4   variety       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
dt['variety'].unique()
```

```
array(['Setosa', 'Versicolor', 'Virginica'],
dtype=object)
```

```
from sklearn.preprocessing import LabelEncoder
```

```
l = LabelEncoder()
```

```
dt['variety'] = l.fit_transform(dt['variety'])
```

```
dt['variety'].unique()
```

```
array([0, 1, 2])
```

```
from sklearn.model_selection import train_test_split
```

```
x = dt.drop(['variety'],axis =1)
y = dt['variety']
```

```
x
```

	sepal.length	sepal.width	petal.length	petal.
0	5.1	3.5	1.4	
1	4.9	3.0	1.4	
2	4.7	3.2	1.3	
3	4.6	3.1	1.5	
4	5.0	3.6	1.4	
...	...	...	...	
145	6.7	3.0	5.2	
146	6.3	2.5	5.0	
147	6.5	3.0	5.2	
148	6.2	3.4	5.4	
149	5.9	3.0	5.1	

150 rows × 4 columns

Next steps: Generate code with x View recommended plots

```
dt.corr()
```

	sepal.length	sepal.width	petal.length
sepal.length	1.000000	-0.117570	0.871754
sepal.width	-0.117570	1.000000	-0.428440
petal.length	0.871754	-0.428440	1.000000
petal.width	0.817941	-0.366126	0.962865
variety	0.782561	-0.426658	0.949035

```
xtrain, xtest, ytrain, ytest = train_test_split(x,y, tes
```

```
xtest
```

	sepal.length	sepal.width	petal.length	peta
15	5.7	4.4	1.5	
98	5.1	2.5	3.0	
93	5.0	2.3	3.3	
19	5.1	3.8	1.5	
89	5.5	2.5	4.0	
37	4.9	3.6	1.4	
0	5.1	3.5	1.4	
94	5.6	2.7	4.2	
66	5.6	3.0	4.5	
97	6.2	2.9	4.3	
49	5.0	3.3	1.4	
132	6.4	2.8	5.6	
63	6.1	2.9	4.7	
58	6.6	2.9	4.6	
90	5.5	2.6	4.4	
142	5.8	2.7	5.1	
70	5.9	3.2	4.8	
53	5.5	2.3	4.0	
84	5.4	3.0	4.5	
62	6.0	2.2	4.0	
33	5.5	4.2	1.4	
122	7.7	2.8	6.7	
76	6.8	2.8	4.8	
77	6.7	3.0	5.0	
32	5.2	4.1	1.5	
123	6.3	2.7	4.9	
82	5.8	2.7	3.9	
136	6.3	3.4	5.6	
114	5.8	2.8	5.1	
3	4.6	3.1	1.5	