```
pip install sdv
```

```
Requirement already satisfied: sdv in /usr/local/lib/python3.10/dist-packages (1.15.0)
Requirement already satisfied: boto3>=1.28 in /usr/local/lib/python3.10/dist-packages (from sdv) (1.34.149)
Requirement already satisfied: botocore>=1.31 in /usr/local/lib/python3.10/dist-packages (from sdv) (1.34.149)
Requirement already satisfied: cloudpickle>=2.1.0 in /usr/local/lib/python3.10/dist-packages (from sdv) (2.2.1)
Requirement already satisfied: graphviz>=0.13.2 in /usr/local/lib/python3.10/dist-packages (from sdv) (0.20.3)
Requirement already satisfied: tqdm>=4.29 in /usr/local/lib/python3.10/dist-packages (from sdv) (4.66.4)
Requirement already satisfied: copulas>=0.11.0 in /usr/local/lib/python3.10/dist-packages (from sdv) (0.11.0)
Requirement already satisfied: ctgan>=0.10.0 in /usr/local/lib/python3.10/dist-packages (from sdv) (0.10.1)
Requirement already satisfied: deepecho>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from sdv) (0.6.0)
Requirement already satisfied: rdt>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from sdv) (1.12.2)
Requirement already satisfied: sdmetrics>=0.14.0 in /usr/local/lib/python3.10/dist-packages (from sdv) (0.15.0)
Requirement already satisfied: platformdirs>=4.0 in /usr/local/lib/python3.10/dist-packages (from sdv) (4.2.2)
Requirement already satisfied: pyyaml>=6.0.1 in /usr/local/lib/python3.10/dist-packages (from sdv) (6.0.1)
Requirement already satisfied: pandas>=1.4.0 in /usr/local/lib/python3.10/dist-packages (from sdv) (2.0.3)
Requirement already satisfied: numpy<2.0.0,>=1.23.3 in /usr/local/lib/python3.10/dist-packages (from sdv) (1.25.2)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from boto3>=1.28->sdv) (1.0.1)
Requirement already satisfied: s3transfer<0.11.0,>=0.10.0 in /usr/local/lib/python3.10/dist-packages (from boto3>=1.28->sdv) (0.10.
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.10/dist-packages (from botocore>=1.31->sdv) (2
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in /usr/local/lib/python3.10/dist-packages (from botocore>=1.31->sdv) (2.
Requirement already satisfied: plotly>=5.10.0 in /usr/local/lib/python3.10/dist-packages (from copulas>=0.11.0->sdv) (5.23.0)
Requirement already satisfied: scipy>=1.9.2 in /usr/local/lib/python3.10/dist-packages (from copulas>=0.11.0->sdv) (1.13.1)
Requirement already satisfied: torch>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from ctgan>=0.10.0->sdv) (2.3.1+cu121)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.4.0->sdv) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.4.0->sdv) (2024.1)
Requirement already satisfied: Faker>=17 in /usr/local/lib/python3.10/dist-packages (from rdt>=1.12.0->sdv) (26.0.0)
Requirement already satisfied: scikit-learn>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from rdt>=1.12.0->sdv) (1.3.2)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly>=5.10.0->copulas>=0.11.0->sd
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from plotly>=5.10.0->copulas>=0.11.0->sdv) (24
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil<3.0.0,>=2.1->botocore>=1.3
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.1.0->rdt>=1.12.0->sdv
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.1.0->rdt>=1.12
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=0.10.0->sdv) (3.15.4
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=0.10
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=0.10.0->sdv) (1.13.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=0.10.0->sdv) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=0.10.0->sdv) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=0.10.0->sdv) (2024.6.1
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctg
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->c
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctg
Requirement already satisfied: nvidia-cudnn-cu12==8.9.2.26 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=0
Requirement already satisfied: nvidia-cublas-cu12==12.1.3.1 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=
Requirement already satisfied: nvidia-cufft-cu12==11.0.2.54 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctg
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctg
Requirement already satisfied: nvidia-nccl-cu12==2.20.5 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=0.10
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=0.
Requirement already satisfied: triton==2.3.1 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=0.10.0->sdv) (2
Requirement already satisfied: nvidia-nvjitlink-cu12 in /usr/local/lib/python3.10/dist-packages (from nvidia-cusolver-cu12==11.4.5.
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.11.0->ctgan>=0.10.
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.11.0->ctgan>=0.1
```

```
pip install table_evaluator
```

```
Requirement already satisfied: table_evaluator in /usr/local/lib/python3.10/dist-packages (1.6.1)
Requirement already satisfied: pandas==2.0.* in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (2.0.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (1.25.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (4.66.4)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (5.9.5)
Requirement already satisfied: dython==0.7.3 in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (0.7.3)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (0.13.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (3.7.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (1.3.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (1.13.1)
Requirement already satisfied: scikit-plot>=0.3.7 in /usr/local/lib/python3.10/dist-packages (from dython==0.7.3->table_evaluator)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas==2.0.*->table_evaluat
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas==2.0.*->table_evaluator) (2024.
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas==2.0.*->table_evaluator) (202
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->table_evaluator) (1.2.
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->table_evaluator) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->table_evaluator) (4.5
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->table_evaluator) (1.4
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->table_evaluator) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->table_evaluator) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->table_evaluator) (3.1.
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->table_evaluator) (1.4.2
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->table_evaluator)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas==2.0.*->tab
import pandas as pd

from sdmetrics.reports.single_table import QualityReport
from ctgan import CTGAN

from rdt import HyperTransformer
```

```
real_data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/CSV_FILE/Iris.csv")
```

```
df = pd.DataFrame(real_data)

print(df.columns)

print("Original DataFrame:")
print(df)
```

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
Original DataFrame:
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
0      1            5.1           3.5            1.4           0.2
1      2            4.9           3.0            1.4           0.2
2      3            4.7           3.2            1.3           0.2
3      4            4.6           3.1            1.5           0.2
4      5            5.0           3.6            1.4           0.2
..   ...            ...           ...            ...           ...
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            Species
0       Iris-setosa
1       Iris-setosa
2       Iris-setosa
3       Iris-setosa
4       Iris-setosa
..              ...
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica

[150 rows x 6 columns]
```

```
NUM_ROWS = 100
NUM_EPOCHS = 1000
BATCH_SIZE = 500
```

```
df.shape
```

```
(150, 6)
```

```
ht = HyperTransformer()
ht.detect_initial_config(data = df)
detected_config = ht.get_config()
display(detected_config)
```

```
{
    "sdtypes": {
        "Id": "numerical",
        "SepalLengthCm": "numerical",
        "SepalWidthCm": "numerical",
        "PetalLengthCm": "numerical",
        "PetalWidthCm": "numerical",
        "Species": "categorical"
    },
    "transformers": {
        "Id": FloatFormatter(),
        "SepalLengthCm": FloatFormatter(),
        "SepalWidthCm": FloatFormatter(),
        "PetalLengthCm": FloatFormatter(),
        "PetalWidthCm": FloatFormatter(),
        "Species": UniformEncoder()
    }
}
```

```
ht.fit(df)
transformed_df = ht.transform(df)
transformed_df
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1.0	5.1	3.5	1.4	0.2	0.164399
1	2.0	4.9	3.0	1.4	0.2	0.028452
2	3.0	4.7	3.2	1.3	0.2	0.031254
3	4.0	4.6	3.1	1.5	0.2	0.282119
4	5.0	5.0	3.6	1.4	0.2	0.225274
...	...	...	...	...	...	...
145	146.0	6.7	3.0	5.2	2.3	0.748878
146	147.0	6.3	2.5	5.0	1.9	0.967718
147	148.0	6.5	3.0	5.2	2.0	0.861117
148	149.0	6.2	3.4	5.4	2.3	0.932372
149	150.0	5.9	3.0	5.1	1.8	0.717605

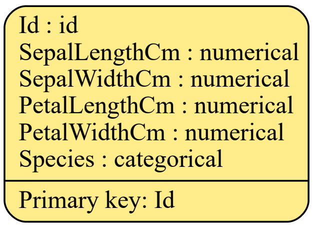
150 rows × 6 columns

Next steps: Generate code with `transformed_df` View recommended plots New interactive sheet

```
import time

start_time = time.time()  # Capture start time before training

model = CTGAN(
    epochs=NUM_EPOCHS,
    verbose=True,
    batch_size=BATCH_SIZE,
    embedding_dim = 1024,
    discriminator_steps = 6,
    discriminator_dim = (512,512)
)

model.fit(transformed_df)

# Training is finished, record end time
end_time = time.time()

# Calculate total training time in seconds
training_time = end_time - start_time

print(f"Training completed! Total time taken: {training_time:.2f} seconds")

model.save("/content/drive/MyDrive/Colab Notebooks/CSV_FILE/Models/iris_1000epochs_500BS_1024_6_512.pkl")
```

```
Gen. (-0.01) | Discrim. (-0.68): 100%|██████████| 1000/1000 [09:55<00:00,  1.68it/s]Training completed! Total time taken: 596.53 se
```

```
from sdv.metadata import SingleTableMetadata
metadata = SingleTableMetadata()
metadata.detect_from_dataframe(df)
metadata_dict= metadata.to_dict()
metadata.visualize()
```


Id : id
SepalLengthCm : numerical
SepalWidthCm : numerical
PetalLengthCm : numerical
PetalWidthCm : numerical
Species : categorical
Primary key: Id

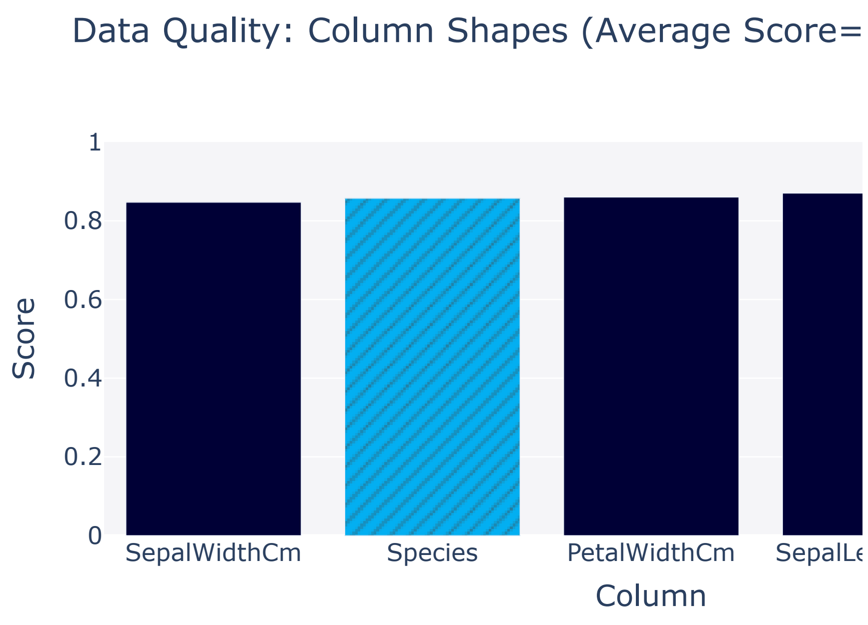
```
categorical_columns = [column for column, info in metadata_dict['columns'].items() if info['sdtype'] == 'categorical']
print(categorical_columns)
```

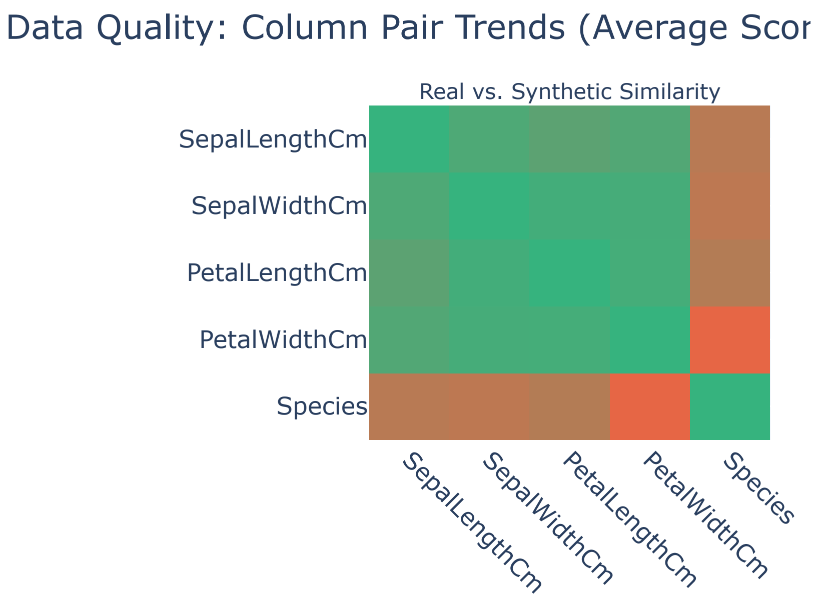
```
['Species']
```

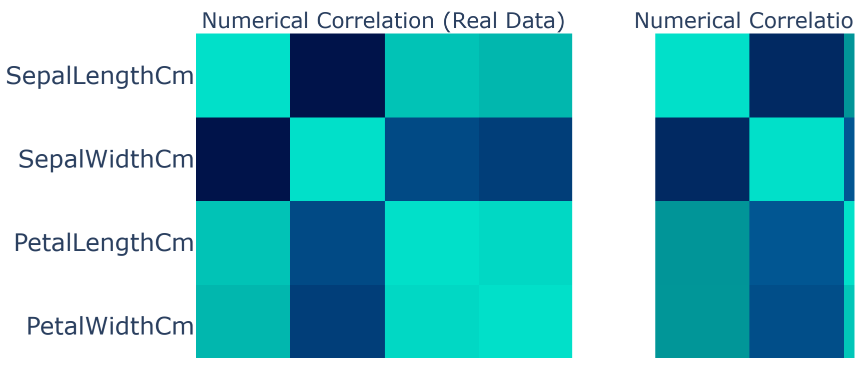
```
from sdmetrics.reports.single_table import QualityReport
```

```
# Get Synthetic data
synthetic_data = model.sample(NUM_ROWS)
# reverse transform the data
synthetic_data = ht.reverse_transform(synthetic_data)

report = QualityReport()
# Use the metadata OBJECT instead of the dictionary
report.generate(df, synthetic_data, metadata.to_dict())

cs_report = report.get_details(property_name="Column Shapes")
print(cs_report)

# Create the first figure
fig1 = report.get_visualization(property_name='Column Shapes')
fig1.show()

# Create the second figure
fig2 = report.get_visualization(property_name='Column Pair Trends')

fig2.show()

report.save(filepath='/content/drive/MyDrive/Colab Notebooks/CSV_FILE/Models/iris_report_1000epochs_500BS_1024_6_512.pkl')
```

```
Generating report ...

(1/2) Evaluating Column Shapes: |██████████| 6/6 [00:00<00:00, 541.53it/s]|
Column Shapes Score: 86.47%

(2/2) Evaluating Column Pair Trends: |██████████| 15/15 [00:00<00:00, 73.58it/s]|
Column Pair Trends Score: 81.31%

Overall Score (Average): 83.89%

          Column        Metric     Score
0  SepalLengthCm  KSComplement  0.870000
1   SepalWidthCm  KSComplement  0.846667
2  PetalLengthCm  KSComplement  0.890000
3   PetalWidthCm  KSComplement  0.860000
4        Species  TVComplement  0.856667
```





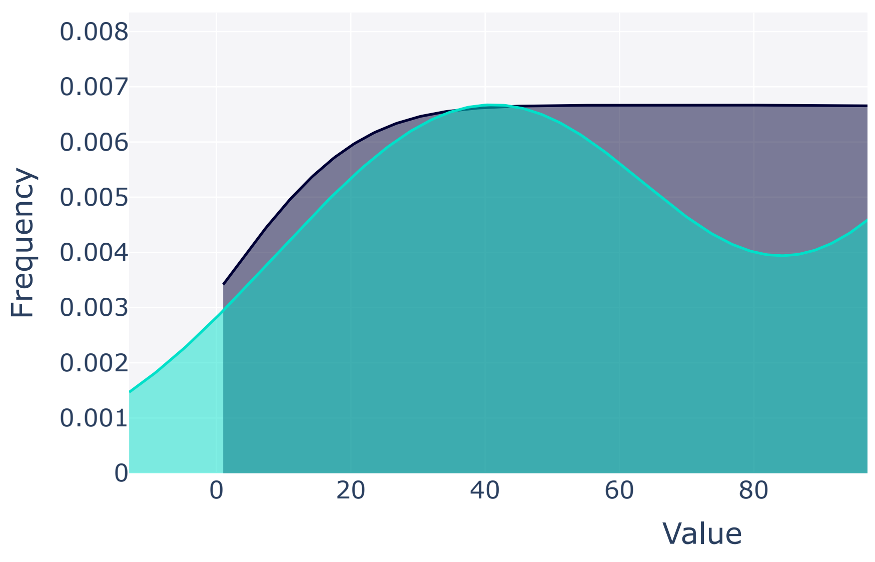


```
from sdmetrics.single_column import CSTest

for column in categorical_columns:
    cstest_result = CSTest.compute(
        real_data=df[column],
        synthetic_data=synthetic_data[column]
    )
    print(f"CSTest for column {column}: {cstest_result}")
```

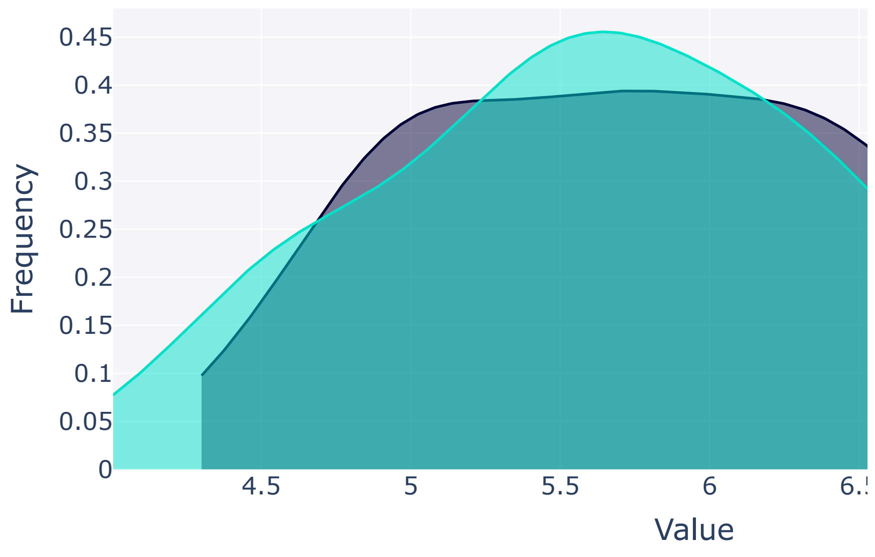
```
CSTest for column Species: 0.9547554925746483
```

```
from sdmetrics.visualization import get_column_plot

# Loop through each column in the dataframe
for column in df.columns:
    fig = get_column_plot(
        real_data=df,
        synthetic_data=synthetic_data,
        column_name=column,
    )

    fig.show()
```

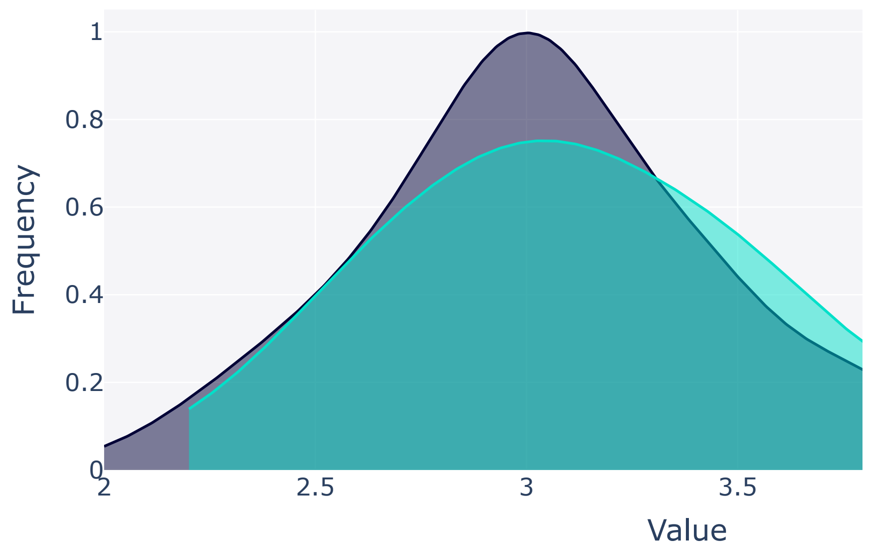
Real vs. Synthetic Data for column 'Id'

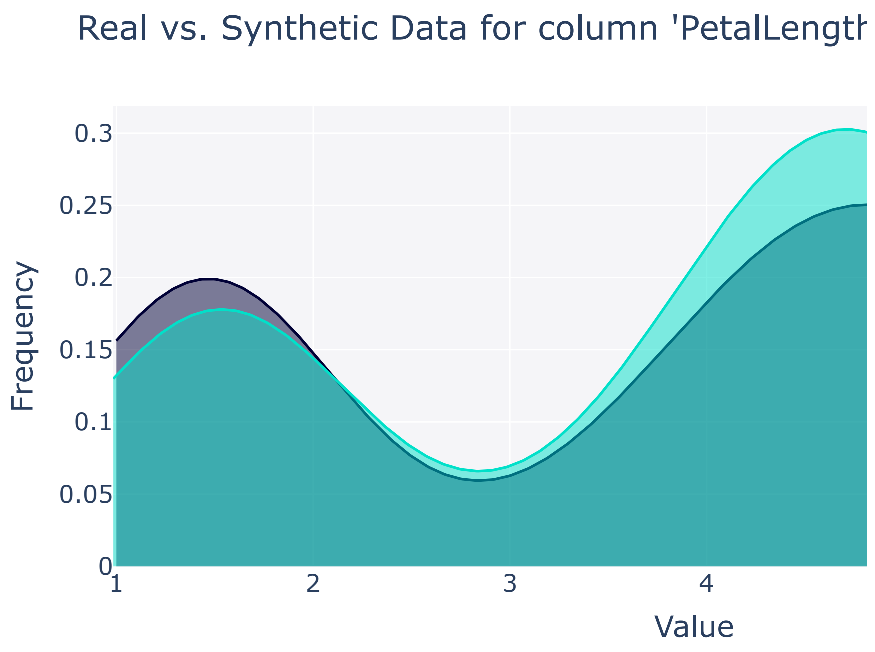


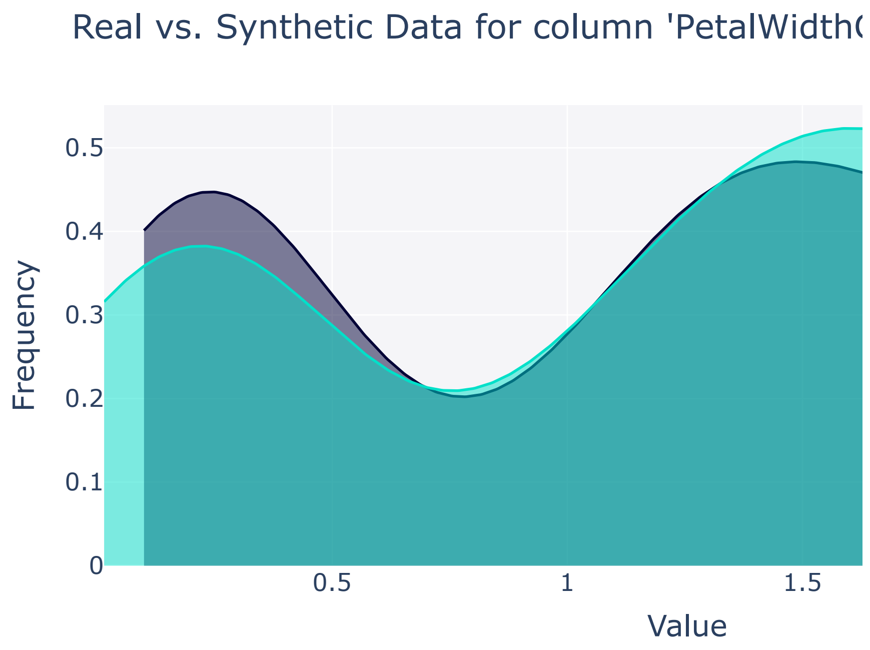
Real vs. Synthetic Data for column 'SepalLengt

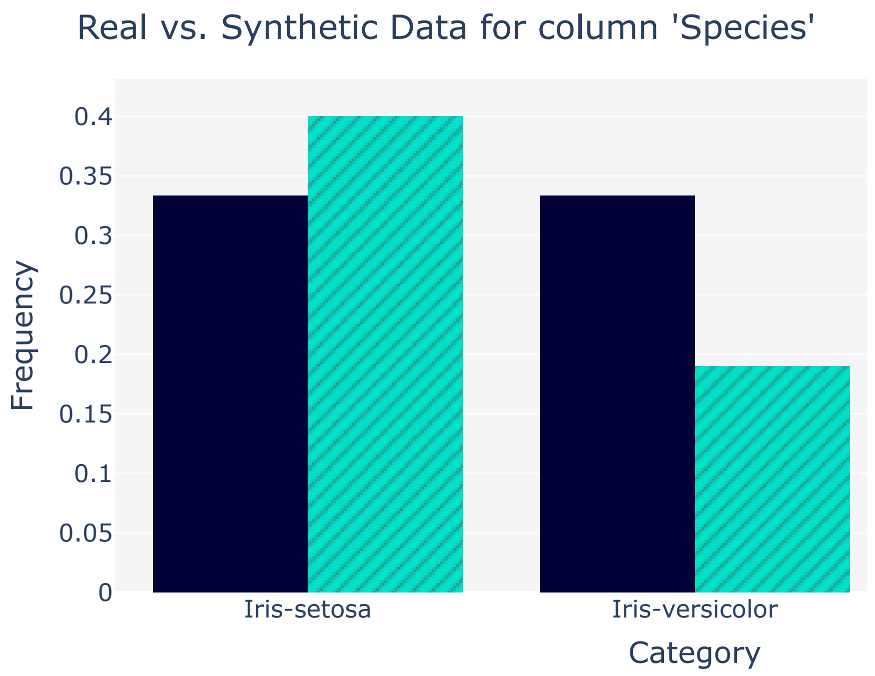


Real vs. Synthetic Data for column 'SepalWidth




Real vs. Synthetic Data for column 'PetalLength
Frequency
0.3
0.25
0.2
0.15
0.1
0.05
0
1
2
3
4
Value


Real vs. Synthetic Data for column 'PetalWidth(
Frequency
0.5
0.4
0.3
0.2
0.1
0
0.5
1
1.5
Value

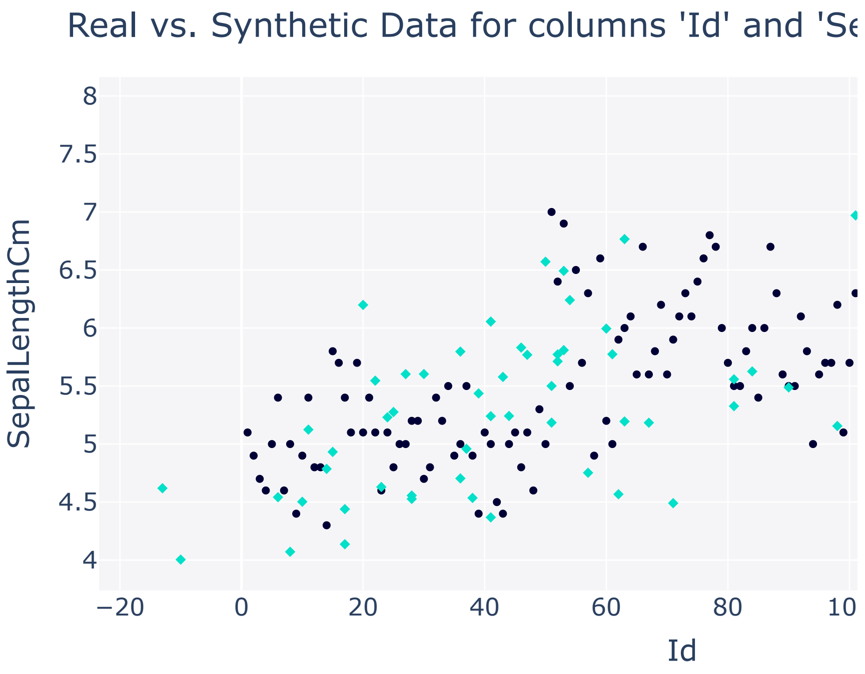

Real vs. Synthetic Data for column 'Species'
Frequency
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Iris-setosa
Iris-versicolor
Category

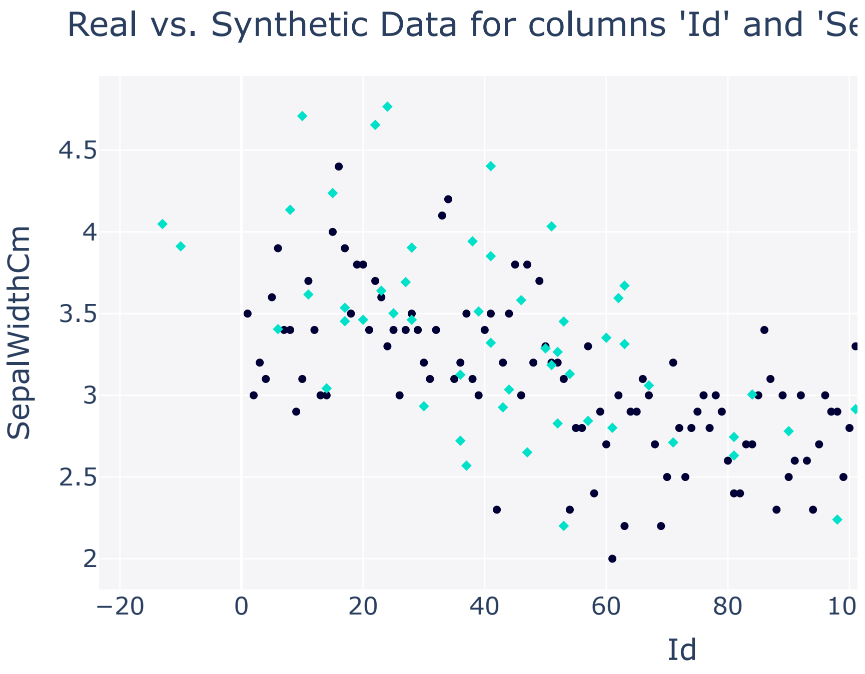
```
display(synthetic_data)
```

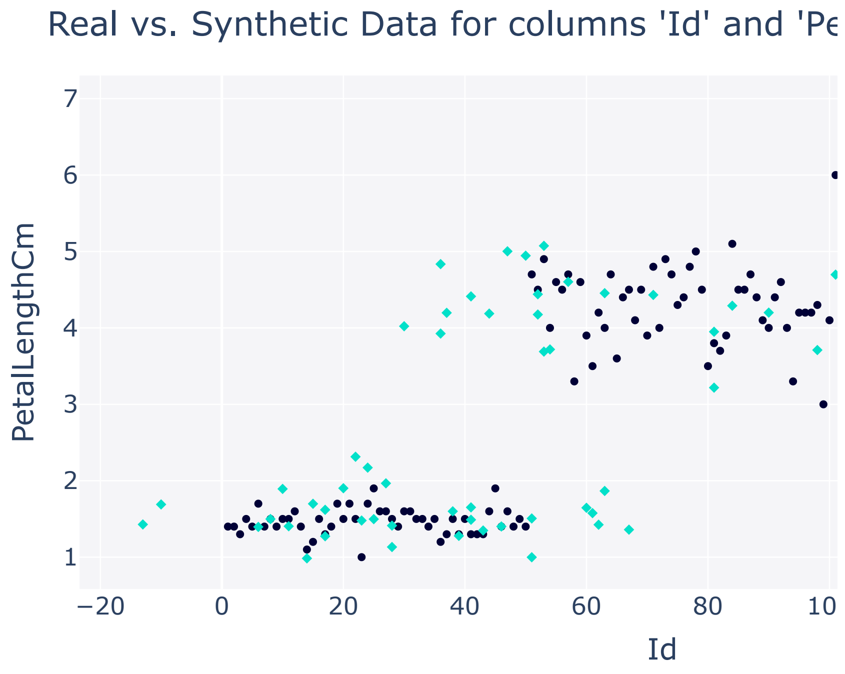
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	104	5.423304	2.884901	4.632228	1.939118	Iris-virginica
1	81	5.326742	2.631638	3.218180	1.249934	Iris-setosa
2	52	5.712609	3.265394	4.440359	1.689444	Iris-setosa
3	106	5.577306	2.367454	4.326265	0.943981	Iris-virginica
4	10	4.502649	4.709066	1.893229	0.432372	Iris-setosa
...	...	...	...	...	...	...
95	131	5.858793	3.038714	4.621559	1.528545	Iris-versicolor
96	14	4.786050	3.042530	0.984551	0.101243	Iris-setosa
97	127	6.039349	3.205812	4.191150	1.689604	Iris-virginica
98	41	6.055466	3.321210	4.412469	1.198574	Iris-versicolor
99	24	5.230453	4.766490	2.171508	0.157332	Iris-versicolor

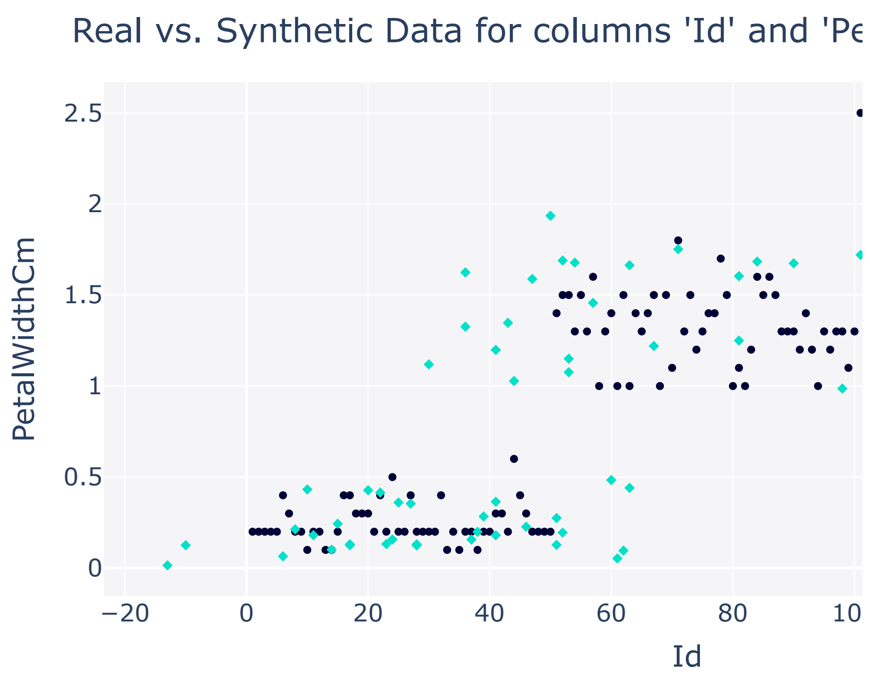
Next steps: Generate code with synthetic_data | View recommended plots | New interactive sheet

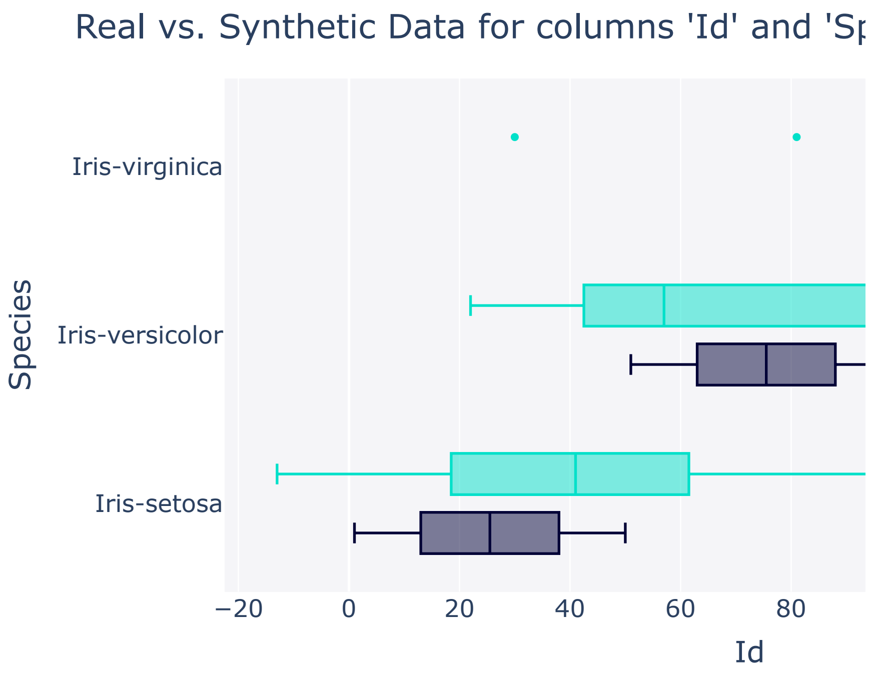
```
from itertools import combinations
from matplotlib.backends.backend_pdf import PdfPages

# Get all column pairs
column_pairs = combinations(df.columns, 2)

# Loop through each column pair
for column1, column2 in column_pairs:
  # Generate the plot using get_column_pair_plot
  fig = get_column_pair_plot(
      real_data=df,
      synthetic_data=synthetic_data,
      column_names=[column1, column2]
  )


  fig.show()
```

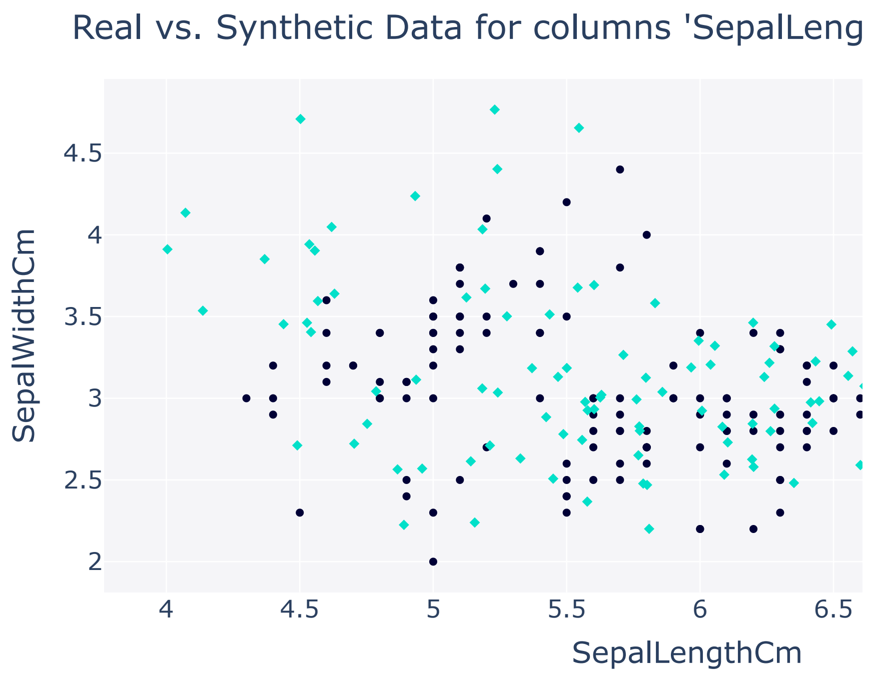

Real vs. Synthetic Data for columns 'Id' and 'Se
SepalLengthCm
Id
−20
0
20
40
60
80
10
4
4.5
5
5.5
6
6.5
7
7.5
8

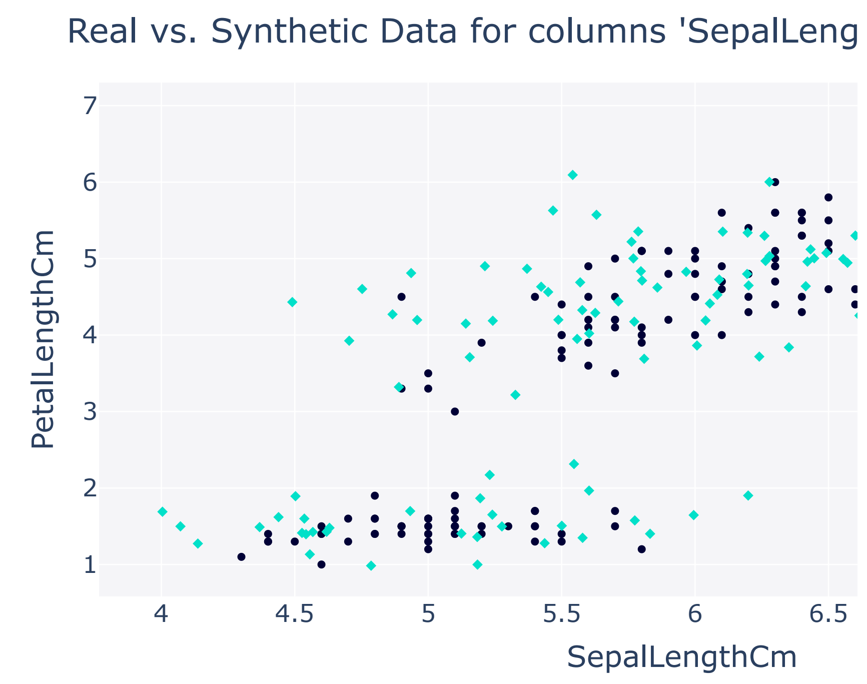

Real vs. Synthetic Data for columns 'Id' and 'Se
SepalWidthCm
Id
−20
0
20
40
60
80
10
2
2.5
3
3.5
4
4.5

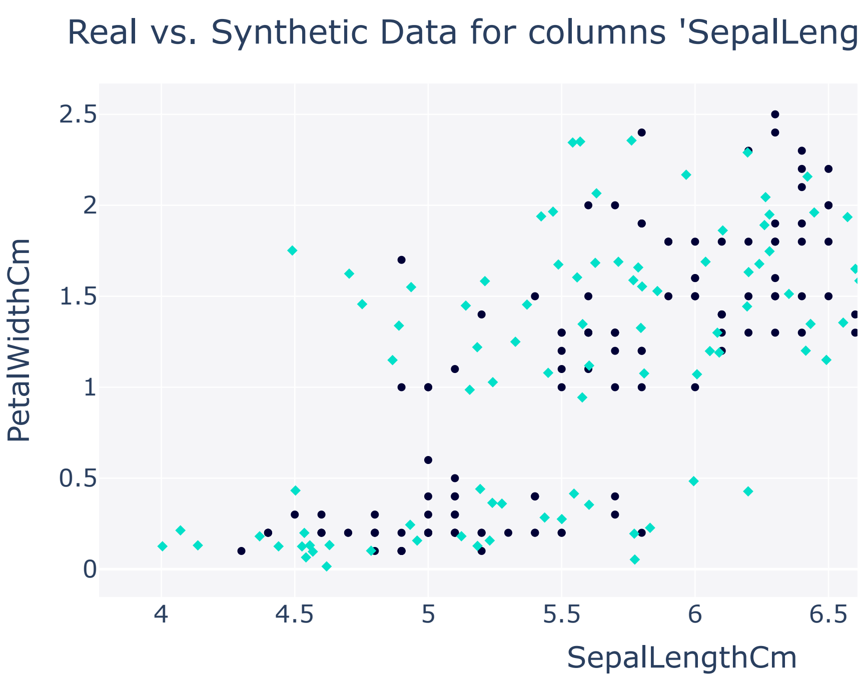

Real vs. Synthetic Data for columns 'Id' and 'Pe
PetalLengthCm
Id
−20
0
20
40
60
80
10
1
2
3
4
5
6
7

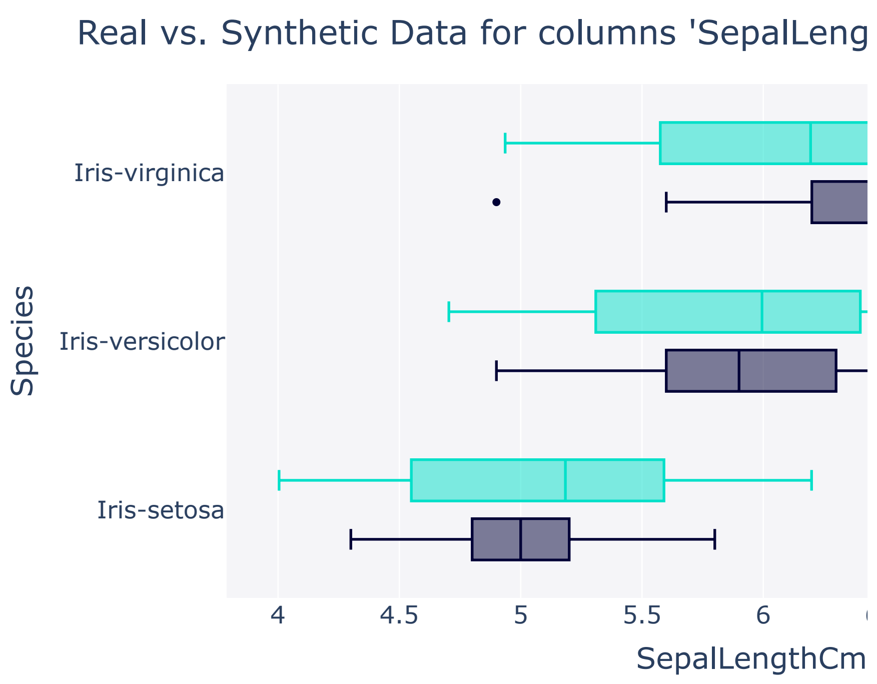

Real vs. Synthetic Data for columns 'Id' and 'Pe
PetalWidthCm
0
0.5
1
1.5
2
2.5
Id
−20
0
20
40
60
80
10

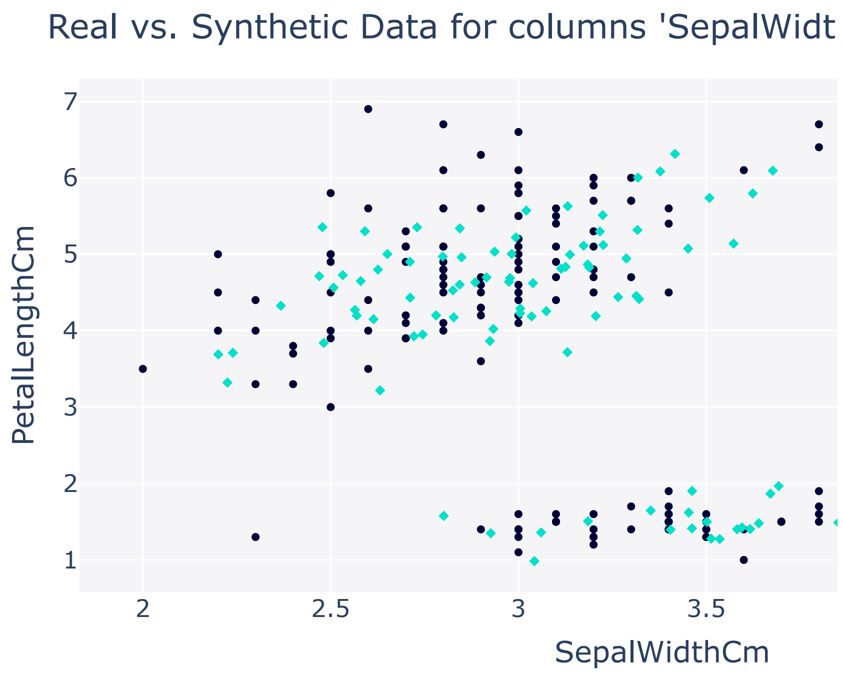

Real vs. Synthetic Data for columns 'Id' and 'Sp
Species
Iris-virginica
Iris-versicolor
Iris-setosa
Id
−20
0
20
40
60
80

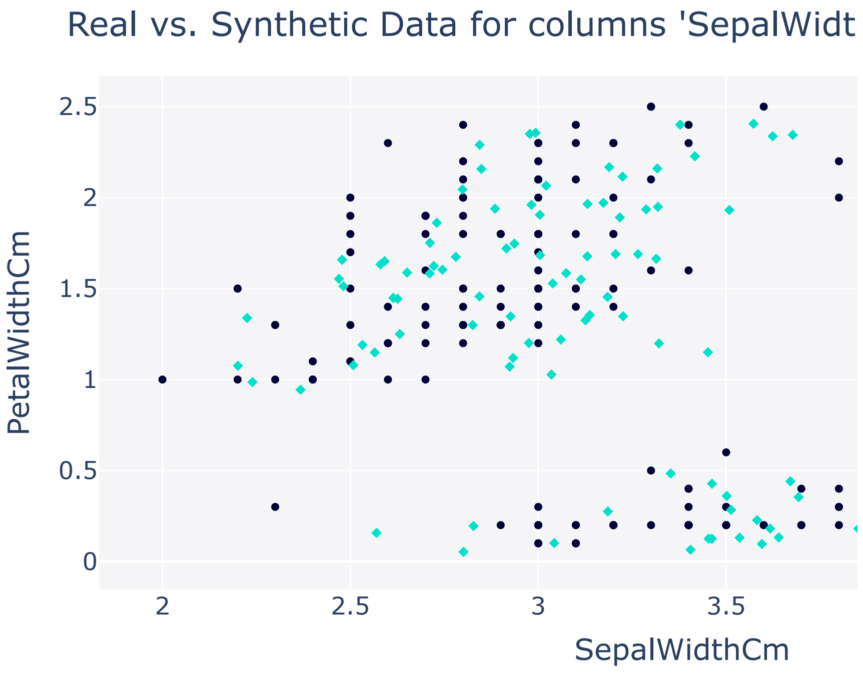

Real vs. Synthetic Data for columns 'SepalLeng
SepalWidthCm
2
2.5
3
3.5
4
4.5
SepalLengthCm
4
4.5
5
5.5
6
6.5

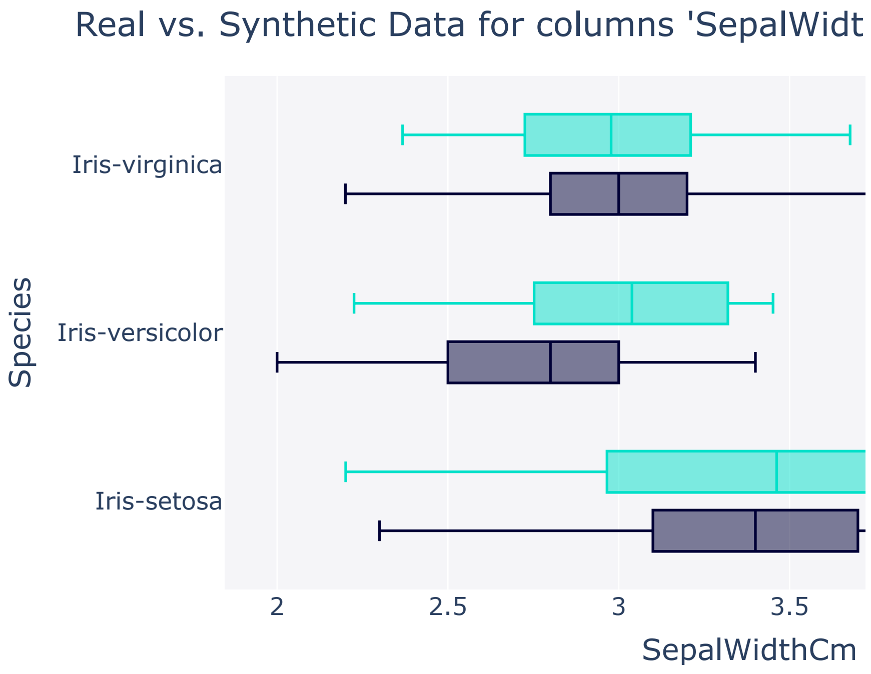

Real vs. Synthetic Data for columns 'SepalLeng
PetalLengthCm
SepalLengthCm
1
2
3
4
5
6
7
4
4.5
5
5.5
6
6.5


Real vs. Synthetic Data for columns 'SepalLeng
PetalWidthCm
SepalLengthCm
0
0.5
1
1.5
2
2.5
4
4.5
5
5.5
6
6.5

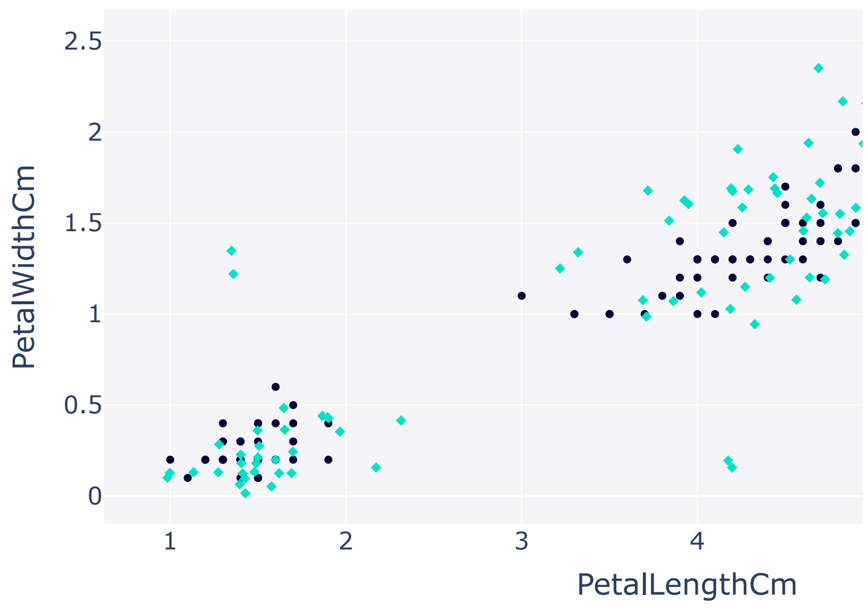

Real vs. Synthetic Data for columns 'SepalLeng
Species
Iris-virginica
Iris-versicolor
Iris-setosa
SepalLengthCm
4
4.5
5
5.5
6


Real vs. Synthetic Data for columns 'SepalWidt
PetalLengthCm
7
6
5
4
3
2
1
2
2.5
3
3.5
SepalWidthCm

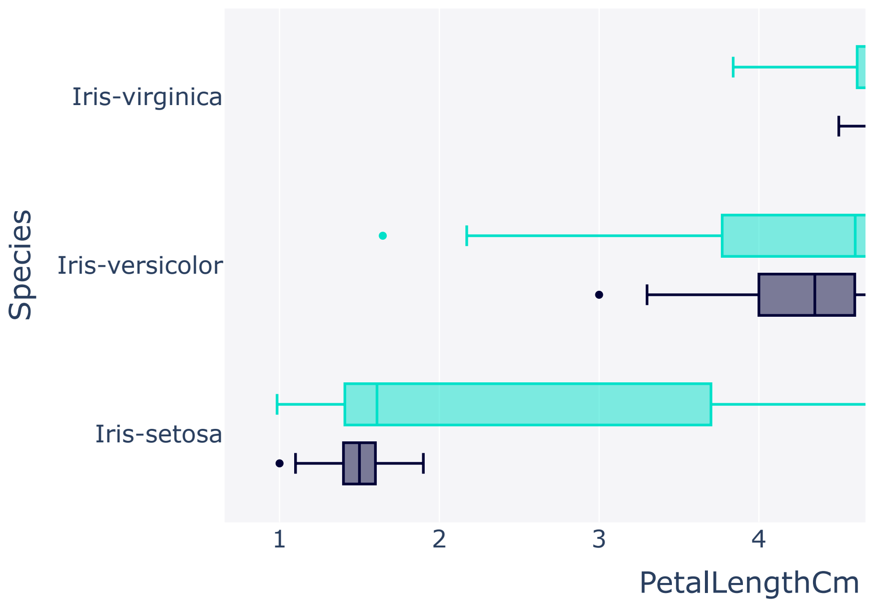

Real vs. Synthetic Data for columns 'SepalWidt
PetalWidthCm
2.5
2
1.5
1
0.5
0
2
2.5
3
3.5
SepalWidthCm


Real vs. Synthetic Data for columns 'SepalWidt
Species
Iris-virginica
Iris-versicolor
Iris-setosa
2
2.5
3
3.5
SepalWidthCm

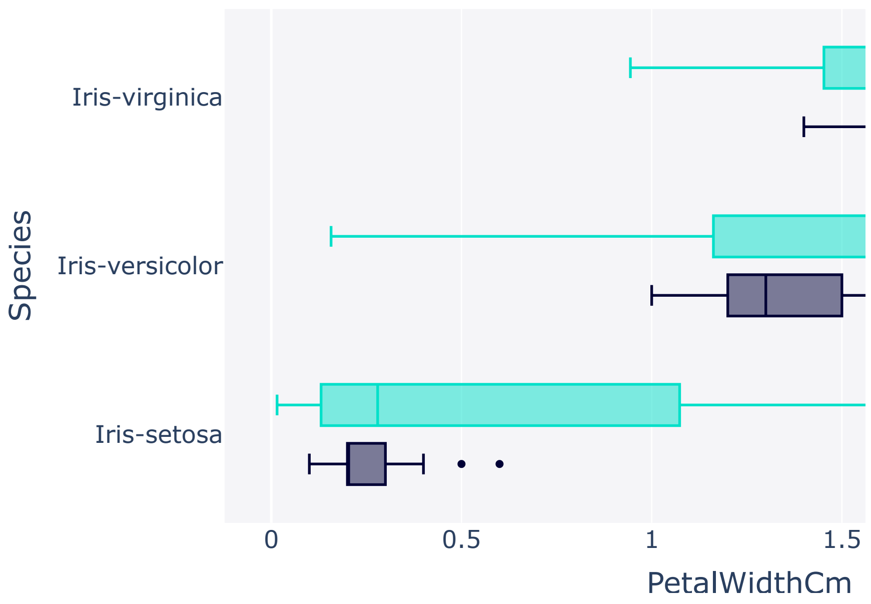
Real vs. Synthetic Data for columns 'PetalLengt



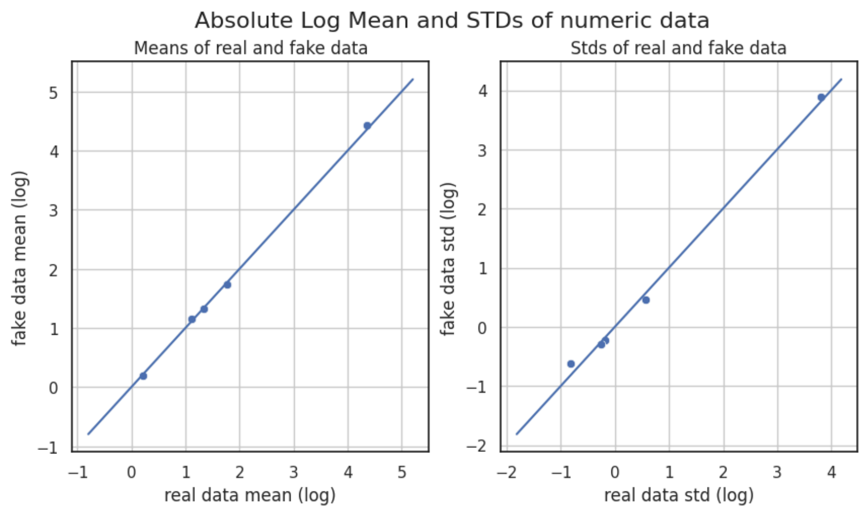
Real vs. Synthetic Data for columns 'PetalLengt

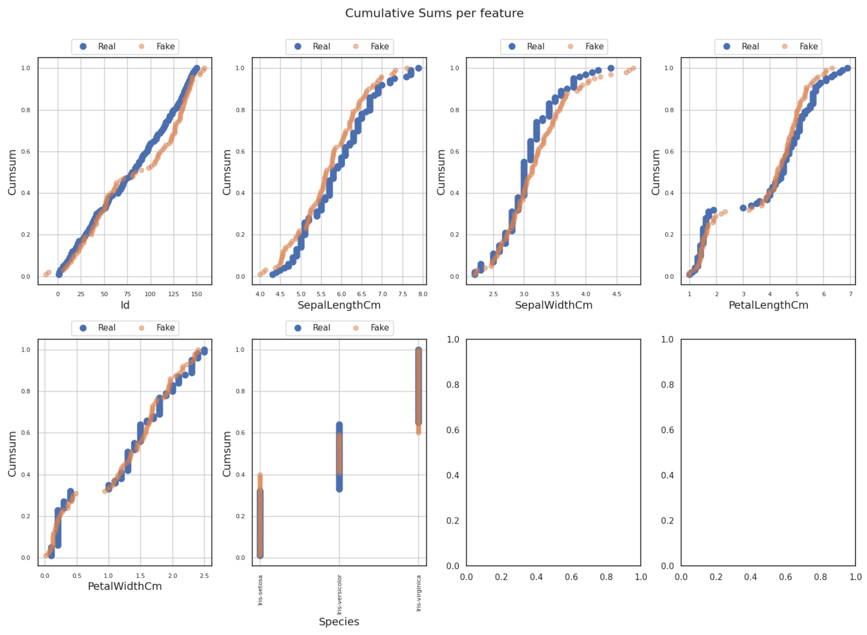


Real vs. Synthetic Data for columns 'PetalWidth

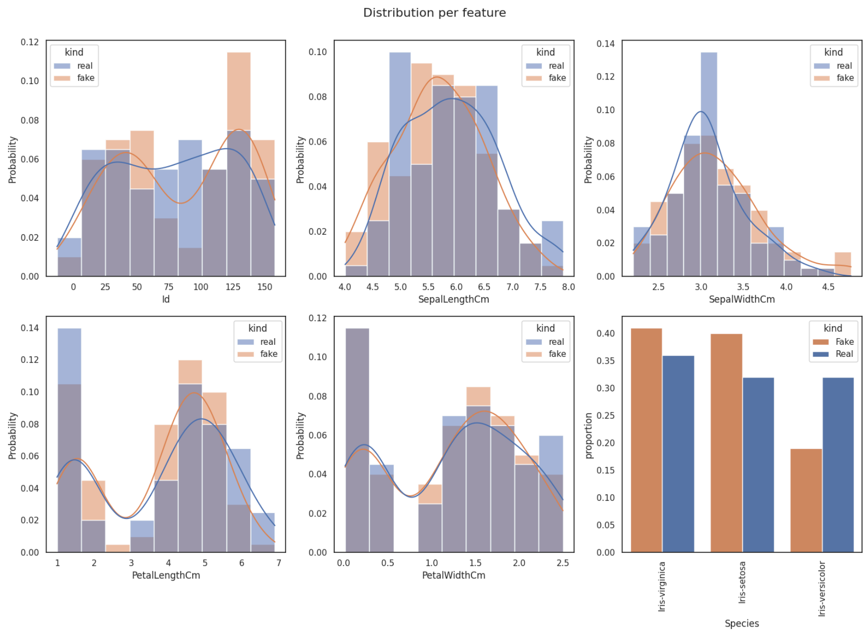


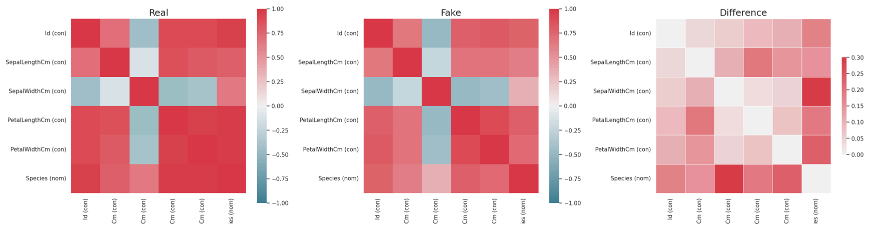
```
from table_evaluator import TableEvaluator

# Assuming real_data and synthetic_data are pandas DataFrames
table_evaluator = TableEvaluator(df, synthetic_data)

table_evaluator.visual_evaluation()
```



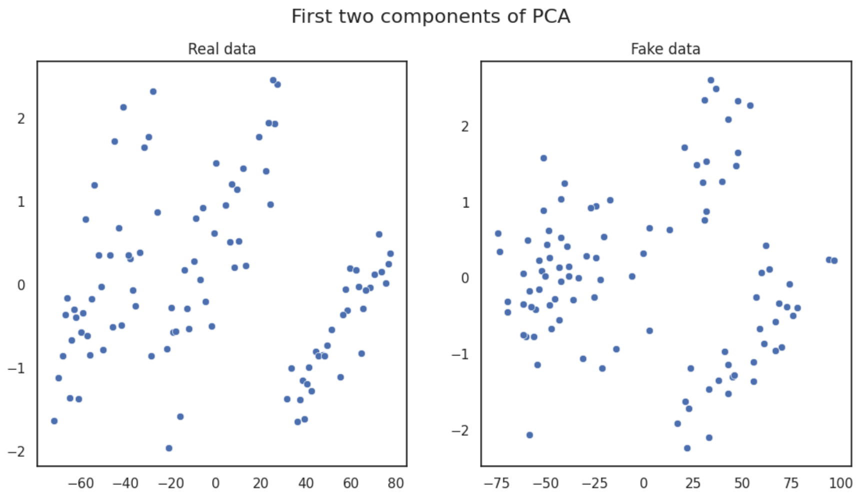


```
/usr/local/lib/python3.10/dist-packages/table_evaluator/table_evaluator.py:182: UserW

FixedFormatter should only be used together with FixedLocator
```





SepalLength
SepalWidth
PetalLength
PetalWidth
Spec
SepalLength
SepalWidth
PetalLength
PetalWidth
Spec
SepalLength
SepalWidth
PetalLength
PetalWidth
Spec


First two components of PCA
Real data
Fake data
2
1
0
−1
−2
−60
−40
−20
0
20
40
60
80
2
1
0
−1
−2
−75
−50
−25
0
25
50
75
100