```
pip install sdv
```

```
Requirement already satisfied: numpy<2.0.0,>=1.23.3 in /usr/local/lib/python3.10/dist-packages (from sdv) (1.26.4)
Collecting jmespath<2.0.0,>=0.7.1 (from boto3<2.0.0,>=1.28->sdv)
  Downloading jmespath-1.0.1-py3-none-any.whl.metadata (7.6 kB)
Collecting s3transfer<0.11.0,>=0.10.0 (from boto3<2.0.0,>=1.28->sdv)
  Downloading s3transfer-0.10.2-py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.10/dist-packages (from botocore<2.0.0,>=1.31->sdv) (2.8.2)
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in /usr/local/lib/python3.10/dist-packages (from botocore<2.0.0,>=1.31->sdv) (2.0.7)
Requirement already satisfied: plotly>=5.10.0 in /usr/local/lib/python3.10/dist-packages (from copulas>=0.11.0->sdv) (5.15.0)
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) (2.4.0+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)
Collecting Faker>=17 (from rdt>=1.12.3->sdv)
  Downloading Faker-28.0.0-py3-none-any.whl.metadata (15 kB)
Requirement already satisfied: scikit-learn>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from rdt>=1.12.3->sdv) (1.3.2)
Collecting plotly>=5.10.0 (from copulas>=0.11.0->sdv)
  Downloading plotly-5.23.0-py3-none-any.whl.metadata (7.3 kB)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly>=5.10.0->copulas>=0.11.0->sdv) (9.0.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from plotly>=5.10.0->copulas>=0.11.0->sdv) (24.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil<3.0.0,>=2.1->botocore<2.0.0,>=1.31->sdv
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.1.0->rdt>=1.12.3->sdv) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.1.0->rdt>=1.12.3->sdv) (3.5
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>=0.10.0->sdv) (4.1
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->ctgan>=0.10.0->sdv) (1.13.2)
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: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.11.0->ctgan>=0.10.0->sdv) (2.1.
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.11.0->ctgan>=0.10.0->sdv) (1.
Downloading sdv-1.16.0-py3-none-any.whl (148 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 148.8/148.8 kB 1.9 MB/s eta 0:00:00
Downloading boto3-1.35.6-py3-none-any.whl (139 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 139.1/139.1 kB 7.5 MB/s eta 0:00:00
Downloading botocore-1.35.6-py3-none-any.whl (12.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.5/12.5 MB 56.5 MB/s eta 0:00:00
Downloading copulas-0.11.1-py3-none-any.whl (51 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 51.6/51.6 kB 3.8 MB/s eta 0:00:00
Downloading ctgan-0.10.1-py3-none-any.whl (24 kB)
Downloading deepecho-0.6.0-py3-none-any.whl (27 kB)
Downloading rdt-1.12.3-py3-none-any.whl (65 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 65.2/65.2 kB 5.0 MB/s eta 0:00:00
Downloading sdmetrics-0.15.1-py3-none-any.whl (170 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 170.7/170.7 kB 11.0 MB/s eta 0:00:00
Downloading Faker-28.0.0-py3-none-any.whl (1.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 61.3 MB/s eta 0:00:00
Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Downloading plotly-5.23.0-py3-none-any.whl (17.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.3/17.3 MB 59.0 MB/s eta 0:00:00
Downloading s3transfer-0.10.2-py3-none-any.whl (82 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 82.7/82.7 kB 6.1 MB/s eta 0:00:00
Installing collected packages: plotly, jmespath, Faker, botocore, s3transfer, rdt, deepecho, copulas, sdmetrics, ctgan, boto3, sdv
  Attempting uninstall: plotly
    Found existing installation: plotly 5.15.0
    Uninstalling plotly-5.15.0:
      Successfully uninstalled plotly-5.15.0
Successfully installed Faker-28.0.0 boto3-1.35.6 botocore-1.35.6 copulas-0.11.1 ctgan-0.10.1 deepecho-0.6.0 jmespath-1.0.1 plotly-5.23.0 rdt-1.1
```

```
pip install table_evaluator
```

```
Collecting table_evaluator
  Downloading table_evaluator-1.6.1-py3-none-any.whl.metadata (8.8 kB)
Collecting pandas==2.0.* (from table_evaluator)
  Downloading pandas-2.0.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (18 kB)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (1.26.4)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (4.66.5)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from table_evaluator) (5.9.5)
Collecting dython==0.7.3 (from table_evaluator)
  Downloading dython-0.7.3-py3-none-any.whl.metadata (3.0 kB)
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)
Collecting scikit-plot>=0.3.7 (from dython==0.7.3->table_evaluator)
  Downloading scikit_plot-0.3.7-py3-none-any.whl.metadata (7.1 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas==2.0.*->table_evaluator) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas==2.0.*->table_evaluator) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas==2.0.*->table_evaluator) (2024.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->table_evaluator) (1.2.1)
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) (4.53.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->table_evaluator) (1.4.5)
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) (3.1.2)
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) (3.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas==2.0.*->table_evaluator) (
Downloading table_evaluator-1.6.1-py3-none-any.whl (22 kB)
Downloading dython-0.7.3-py3-none-any.whl (23 kB)
Downloading pandas-2.0.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.3/12.3 MB 30.1 MB/s eta 0:00:00
Downloading scikit_plot-0.3.7-py3-none-any.whl (33 kB)
Installing collected packages: pandas, scikit-plot, dython, table_evaluator
  Attempting uninstall: pandas
```

```
    Found existing installation: pandas 2.1.4
    Uninstalling pandas-2.1.4:
      Successfully uninstalled pandas-2.1.4
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the fol
google-colab 1.0.0 requires pandas==2.1.4, but you have pandas 2.0.3 which is incompatible.
Successfully installed dython-0.7.3 pandas-2.0.3 scikit-plot-0.3.7 table_evaluator-1.6.1
```

```
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/creditcard.csv")


df = pd.DataFrame(real_data)

print(df.columns)

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

```
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
       'Class'],
      dtype='object')
Original DataFrame:
            Time         V1         V2        V3        V4        V5  \
0            0.0  -1.359807  -0.072781  2.536347  1.378155 -0.338321
1            0.0   1.191857   0.266151  0.166480  0.448154  0.060018
2            1.0  -1.358354  -1.340163  1.773209  0.379780 -0.503198
3            1.0  -0.966272  -0.185226  1.792993 -0.863291 -0.010309
4            2.0  -1.158233   0.877737  1.548718  0.403034 -0.407193
...          ...        ...        ...       ...       ...       ...
284802  172786.0 -11.881118  10.071785 -9.834783 -2.066656 -5.364473
284803  172787.0  -0.732789  -0.055080  2.035030 -0.738589  0.868229
284804  172788.0   1.919565  -0.301254 -3.249640 -0.557828  2.630515
284805  172788.0  -0.240440   0.530483  0.702510  0.689799 -0.377961
284806  172792.0  -0.533413  -0.189733  0.703337 -0.506271 -0.012546

              V6        V7        V8        V9  ...       V21       V22  \
0       0.462388  0.239599  0.098698  0.363787  ... -0.018307  0.277838
1      -0.082361 -0.078803  0.085102 -0.255425  ... -0.225775 -0.638672
2       1.800499  0.791461  0.247676 -1.514654  ...  0.247998  0.771679
3       1.247203  0.237609  0.377436 -1.387024  ... -0.108300  0.005274
4       0.095921  0.592941 -0.270533  0.817739  ... -0.009431  0.798278
...          ...       ...       ...       ...  ...       ...       ...
284802 -2.606837 -4.918215  7.305334  1.914428  ...  0.213454  0.111864
284803  1.058415  0.024330  0.294869  0.584800  ...  0.214205  0.924384
284804  3.031260 -0.296827  0.708417  0.432454  ...  0.232045  0.578229
284805  0.623708 -0.686180  0.679145  0.392087  ...  0.265245  0.800049
284806 -0.649617  1.577006 -0.414650  0.486180  ...  0.261057  0.643078

             V23       V24       V25       V26       V27       V28  Amount  \
0      -0.110474  0.066928  0.128539 -0.189115  0.133558 -0.021053  149.62
1       0.101288 -0.339846  0.167170  0.125895 -0.008983  0.014724    2.69
2       0.909412 -0.689281 -0.327642 -0.139097 -0.055353 -0.059752  378.66
3      -0.190321 -1.175575  0.647376 -0.221929  0.062723  0.061458  123.50
4      -0.137458  0.141267 -0.206010  0.502292  0.219422  0.215153   69.99
...          ...       ...       ...       ...       ...       ...     ...
284802  1.014480 -0.509348  1.436807  0.250034  0.943651  0.823731    0.77
284803  0.012463 -1.016226 -0.606624 -0.395255  0.068472 -0.053527   24.79
284804 -0.037501  0.640134  0.265745 -0.087371  0.004455 -0.026561   67.88
284805 -0.163298  0.123205 -0.569159  0.546668  0.108821  0.104533   10.00
284806  0.376777  0.008797 -0.473649 -0.818267 -0.002415  0.013649  217.00

        Class
0           0
1           0
2           0
3           0
4           0
...       ...
284802      0
284803      0
284804      0
284805      0
284806      0
```

```
NUM_ROWS = 200000
NUM_EPOCHS = 1000
BATCH_SIZE = 1500


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

```
{
    "sdtypes": {
        "Time": "numerical",
        "V1": "numerical",
        "V2": "numerical",
        "V3": "numerical",
        "V4": "numerical",
        "V5": "numerical",
        "V6": "numerical",
        "V7": "numerical",
        "V8": "numerical",
        "V9": "numerical",
        "V10": "numerical",
        "V11": "numerical",
        "V12": "numerical",
        "V13": "numerical",
        "V14": "numerical",
        "V15": "numerical",
        "V16": "numerical",
        "V17": "numerical",
        "V18": "numerical",
        "V19": "numerical",
        "V20": "numerical",
        "V21": "numerical",
        "V22": "numerical",
        "V23": "numerical",
        "V24": "numerical",
        "V25": "numerical",
        "V26": "numerical",
        "V27": "numerical",
        "V28": "numerical",
        "Amount": "numerical",
        "Class": "numerical"
    },
    "transformers": {
        "Time": FloatFormatter(),
        "V1": FloatFormatter(),
        "V2": FloatFormatter(),
        "V3": FloatFormatter(),
        "V4": FloatFormatter(),
        "V5": FloatFormatter(),
        "V6": FloatFormatter(),
        "V7": FloatFormatter(),
        "V8": FloatFormatter(),
        "V9": FloatFormatter(),
        "V10": FloatFormatter(),
        "V11": FloatFormatter(),
        "V12": FloatFormatter(),
        "V13": FloatFormatter(),
        "V14": FloatFormatter(),
        "V15": FloatFormatter(),
        "V16": FloatFormatter(),
        "V17": FloatFormatter(),
        "V18": FloatFormatter(),
        "V19": FloatFormatter(),
        "V20": FloatFormatter(),
        "V21": FloatFormatter(),
        "V22": FloatFormatter(),
        "V23": FloatFormatter(),
        "V24": FloatFormatter(),
        "V25": FloatFormatter(),
        "V26": FloatFormatter(),
        "V27": FloatFormatter(),
        "V28": FloatFormatter(),
        "Amount": FloatFormatter(),
        "Class": FloatFormatter()
    }
}
```


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

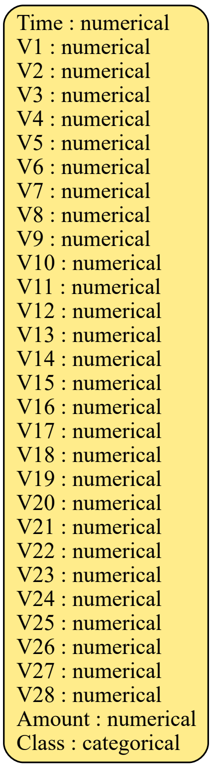
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.800049	-0.163298	0.
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.643078	0.376777	0.

284807 rows × 31 columns

```
import time
import torch as torch

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 = torch.load("/content/drive/MyDrive/Colab Notebooks/credit_1000epcohs_1500BS_1024_6_512_Score98.pkl")
# 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/creditcard_1000epochs_1500BS_1024_6_512.pkl")
```

```
<ipython-input-10-bc9764ba7898>:15: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default value), which uses the
  model = torch.load("/content/drive/MyDrive/Colab Notebooks/credit_1000epcohs_1500BS_1024_6_512_Score98.pkl")
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:348: InconsistentVersionWarning: Trying to unpickle estimator BayesianGaussianMixture from
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
Training completed! Total time taken: 0.80 seconds
```

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




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

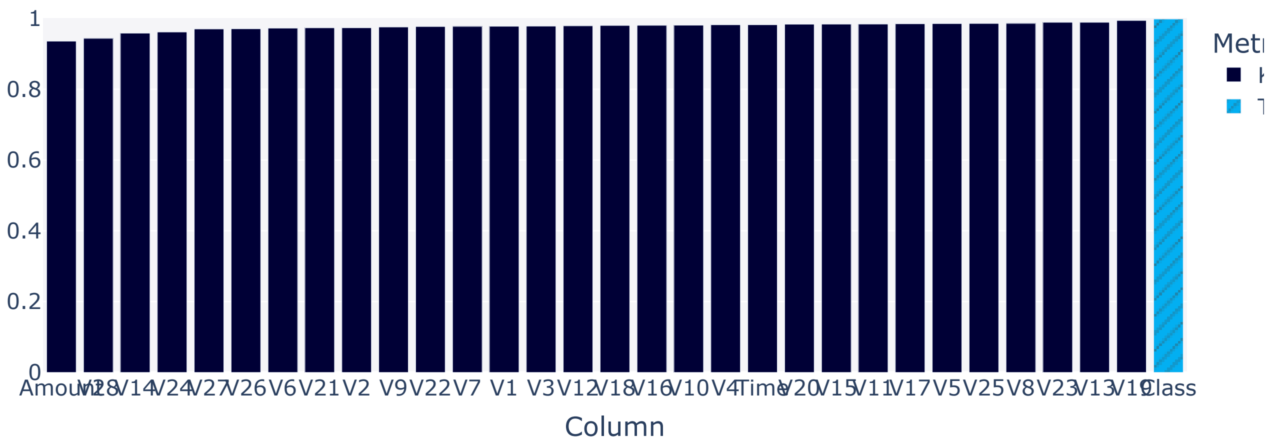
```
['Class']
```

```
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
```

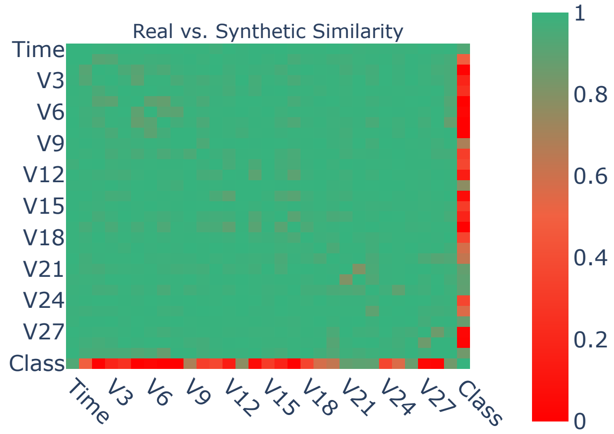
```
report.generate(df, synthetic_data, metadata.to_dict())

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

fig1 = report.get_visualization(property_name='Column Shapes')
#fig1.update_layout(height = 1600,width=1600, margin=dict(l=400, r=400, t=400, b=400))
fig1.show()

# Create the second figure
fig2 = report.get_visualization(property_name='Column Pair Trends')
#fig2.update_layout(height = 1600,width=1600,margin=dict(l=400, r=400, t=400, b=400))
fig2.show()

#print(fig1)

#fig1.write_image("/content/drive/MyDrive/Colab Notebooks/hey.pdf",engine='kaleido')

report.save(filepath='/content/drive/MyDrive/Colab Notebooks/creditcard_report_1000epochs_1500BS_1024_6_512.pkl')
```

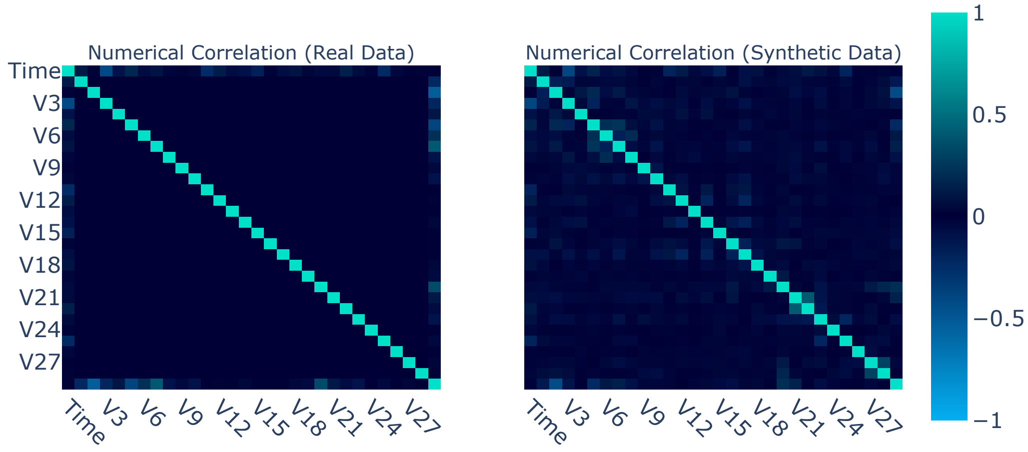
```
Generating report ...

(1/2) Evaluating Column Shapes: |██████████| 31/31 [00:07<00:00,  4.08it/s]|
Column Shapes Score: 97.72%

(2/2) Evaluating Column Pair Trends: |██████████| 465/465 [01:10<00:00,  6.63it/s]|
Column Pair Trends Score: 94.32%

Overall Score (Average): 96.02%

    Column       Metric     Score
0     Time  KSComplement  0.982020
1       V1  KSComplement  0.977771
2       V2  KSComplement  0.973498
3       V3  KSComplement  0.978161
4       V4  KSComplement  0.981967
5       V5  KSComplement  0.985304
6       V6  KSComplement  0.972476
7       V7  KSComplement  0.977240
8       V8  KSComplement  0.985779
9       V9  KSComplement  0.975695
10     V10  KSComplement  0.980828
11     V11  KSComplement  0.984008
12     V12  KSComplement  0.978928
13     V13  KSComplement  0.988929
14     V14  KSComplement  0.958270
15     V15  KSComplement  0.983798
16     V16  KSComplement  0.980418
17     V17  KSComplement  0.984791
18     V18  KSComplement  0.980035
19     V19  KSComplement  0.994095
20     V20  KSComplement  0.983559
21     V21  KSComplement  0.973351
22     V22  KSComplement  0.977027
23     V23  KSComplement  0.988895
24     V24  KSComplement  0.961546
25     V25  KSComplement  0.985655
26     V26  KSComplement  0.970639
27     V27  KSComplement  0.970122
28     V28  KSComplement  0.943452
29  Amount  KSComplement  0.935690
30   Class  TVComplement  0.998273
```

Data Quality: Column Shapes (Average Score=0.98)



Data Quality: Column Pair Trends (Average Score=0.94)



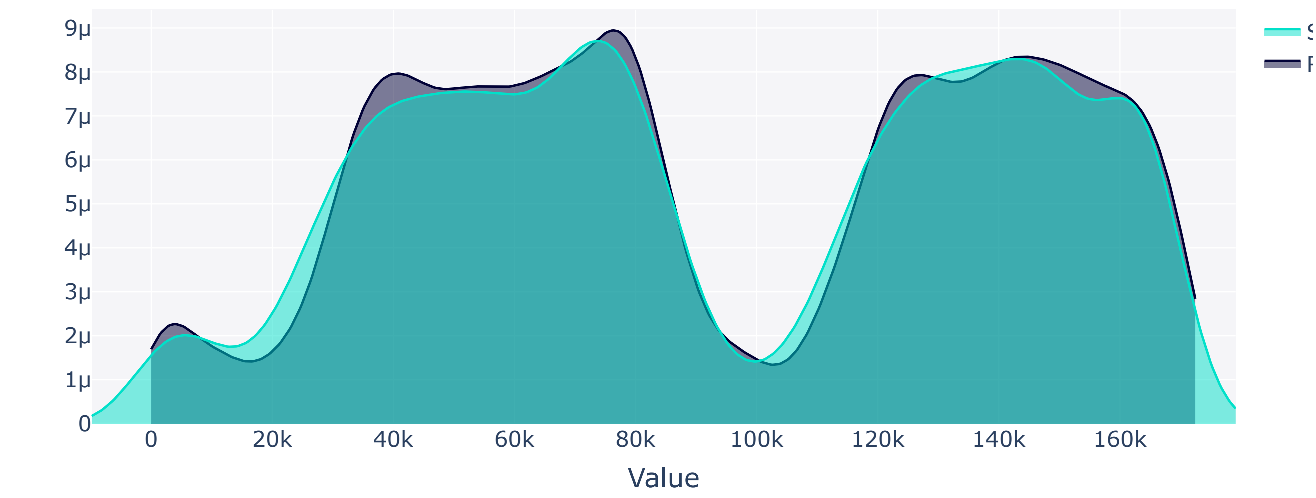



```
from sdmetrics.single_column import CSTest

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}")
```

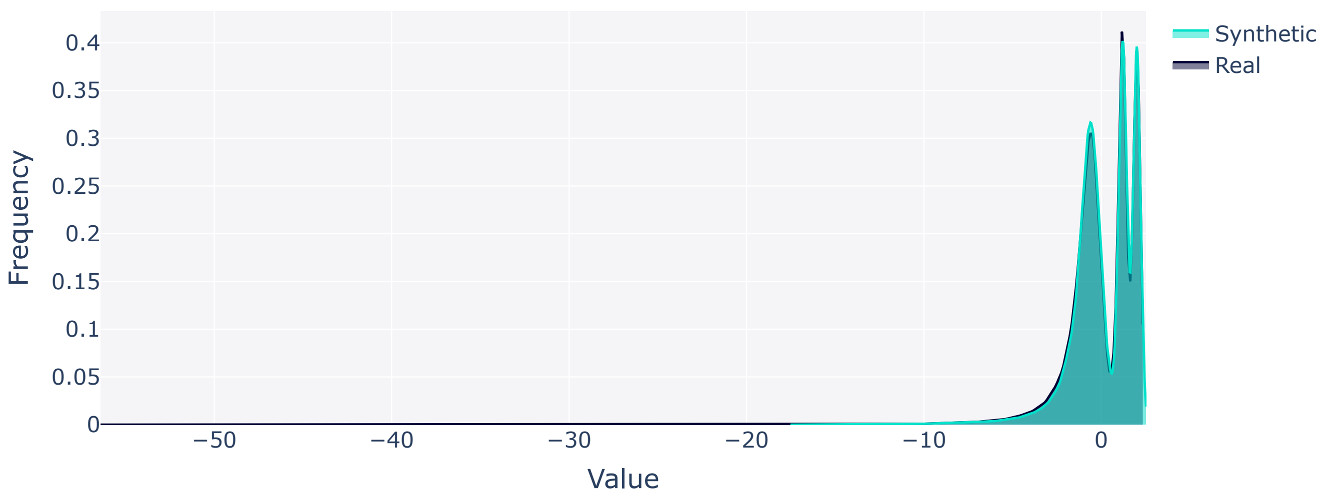
```
CSTest for column Class: 0.9668183876821032
```

```
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()
```

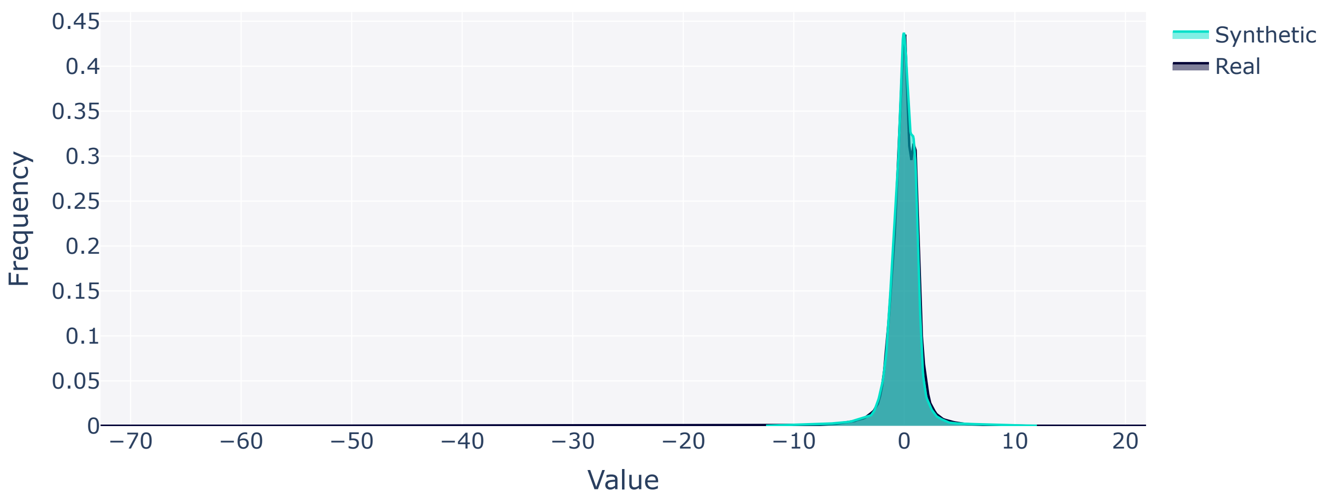
Real vs. Synthetic Data for column 'Time'



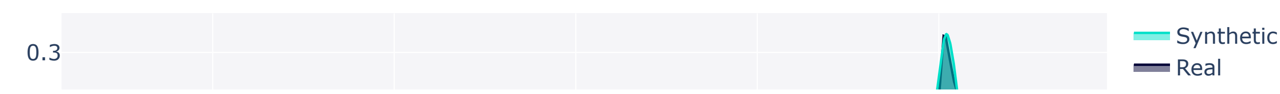
Real vs. Synthetic Data for column 'V1'

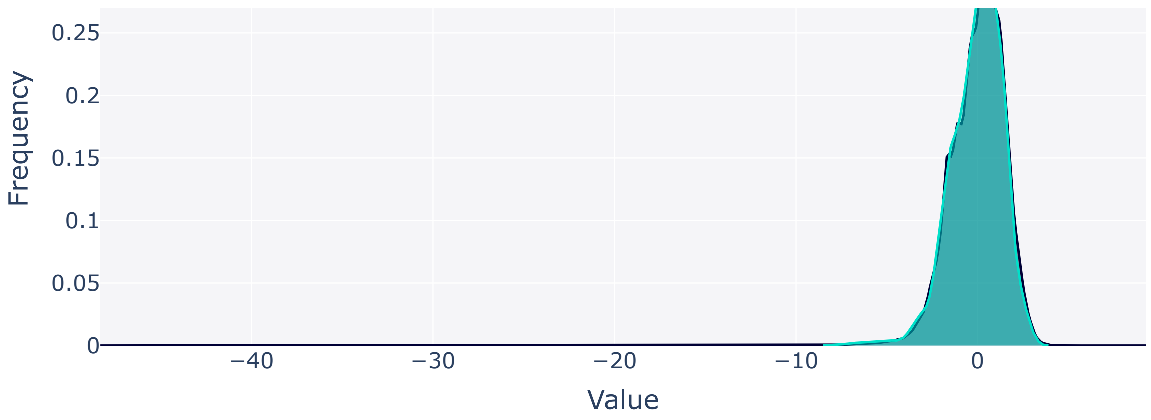


Real vs. Synthetic Data for column 'V2'

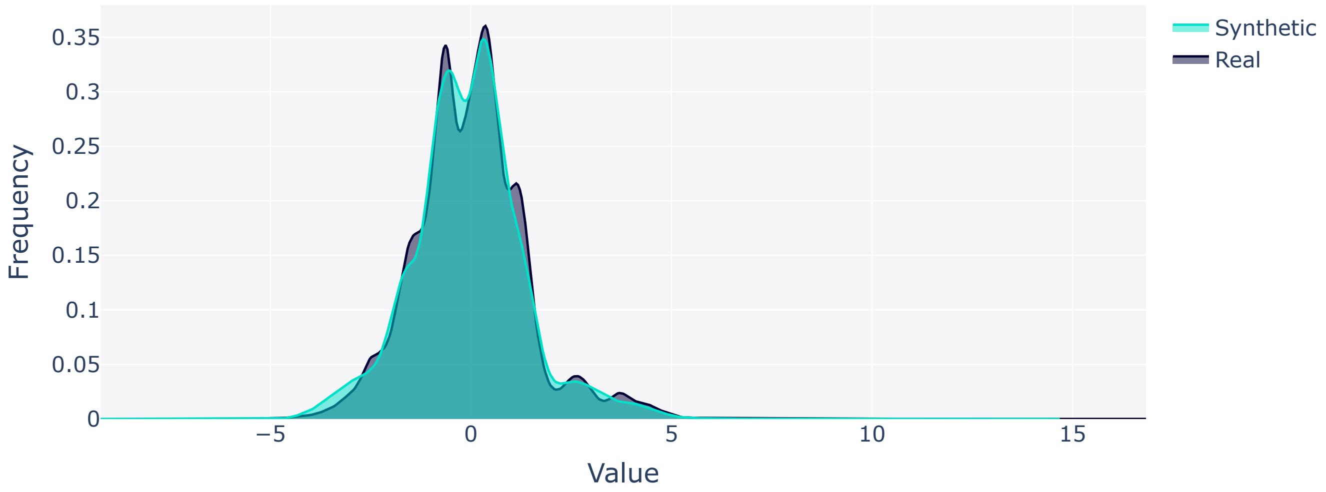


Real vs. Synthetic Data for column 'V3'

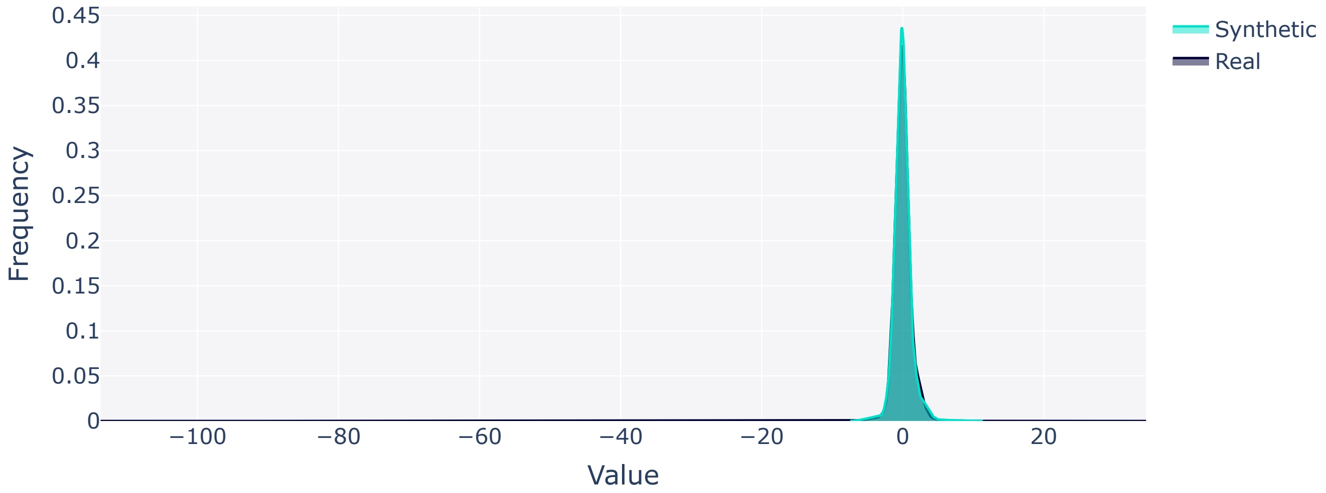




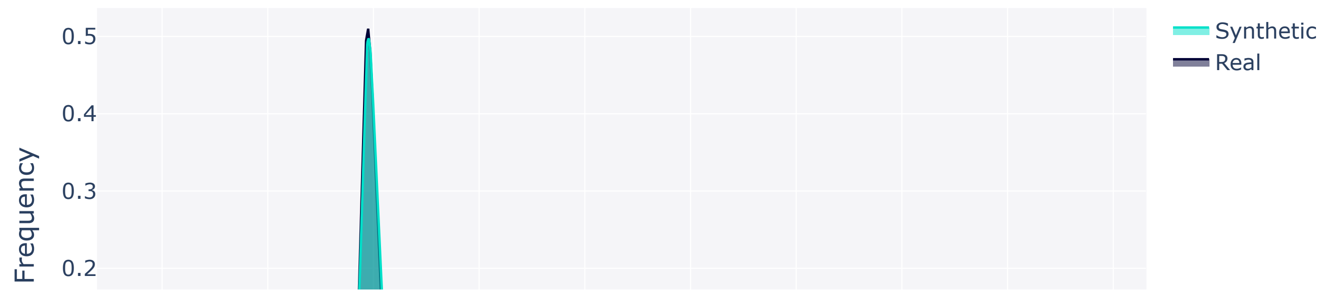
Real vs. Synthetic Data for column 'V4'

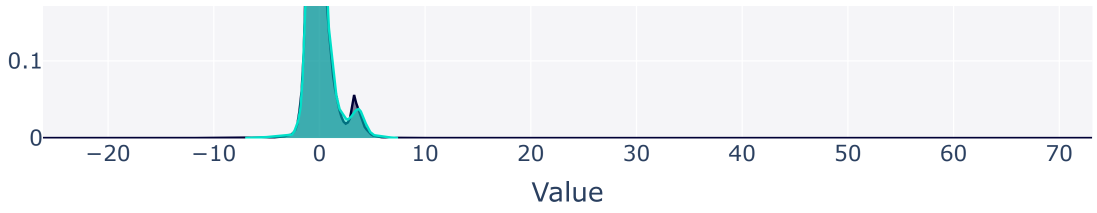


Real vs. Synthetic Data for column 'V5'

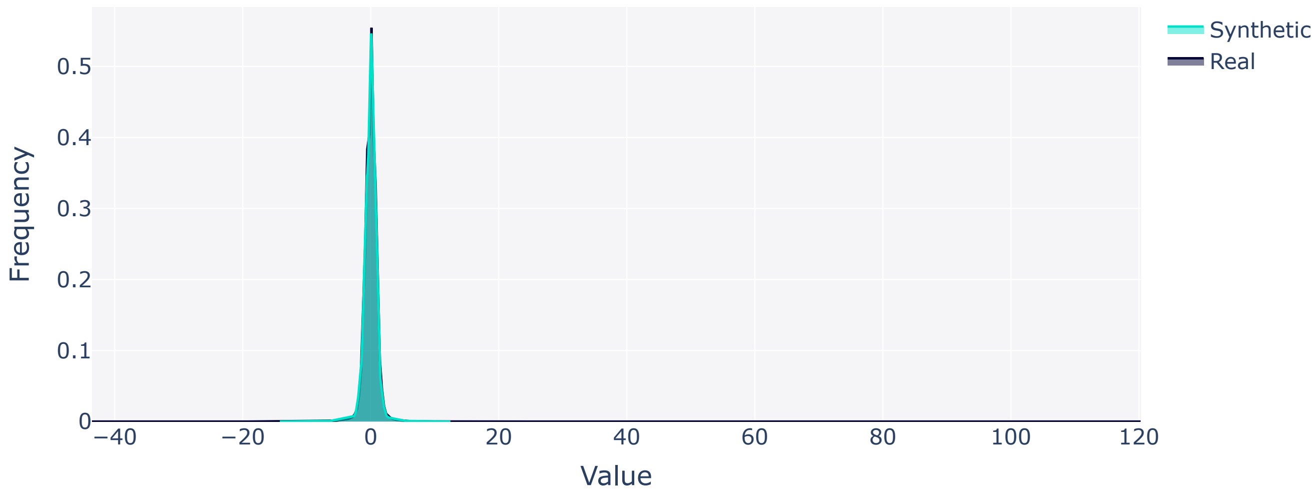


Real vs. Synthetic Data for column 'V6'

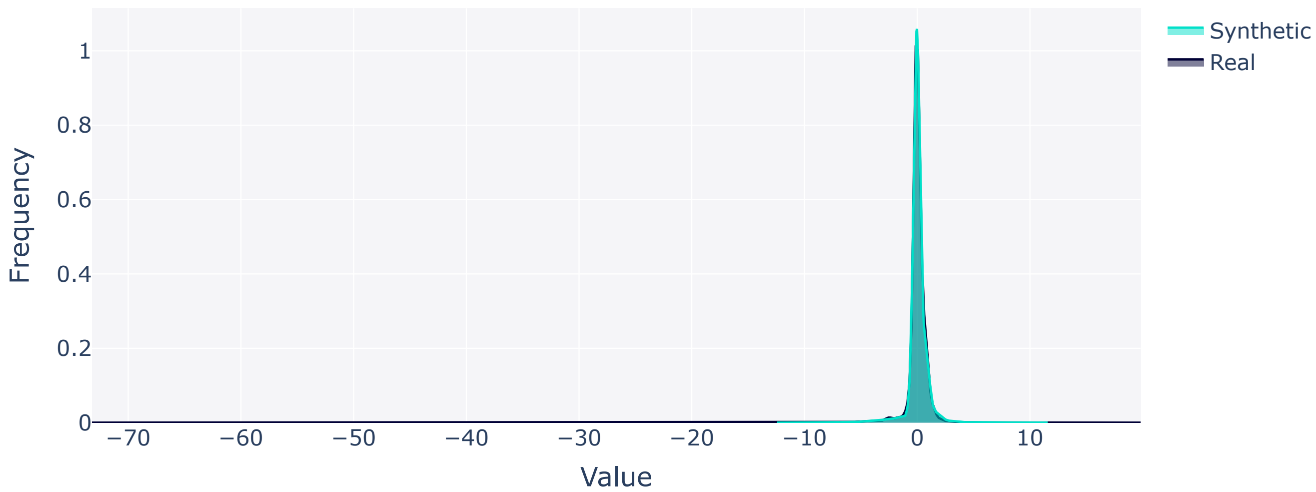




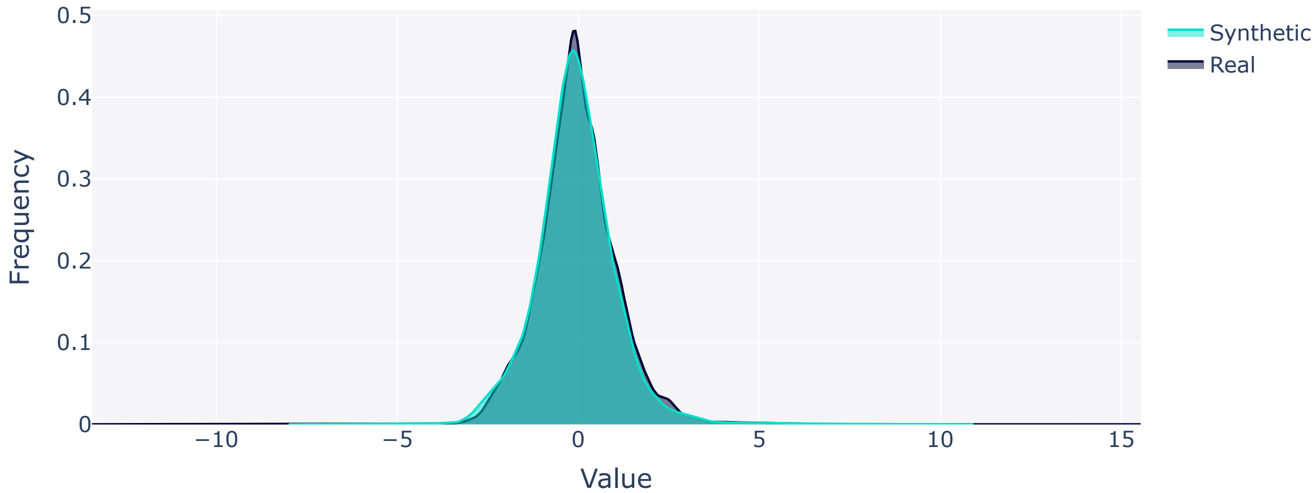
Real vs. Synthetic Data for column 'V7'



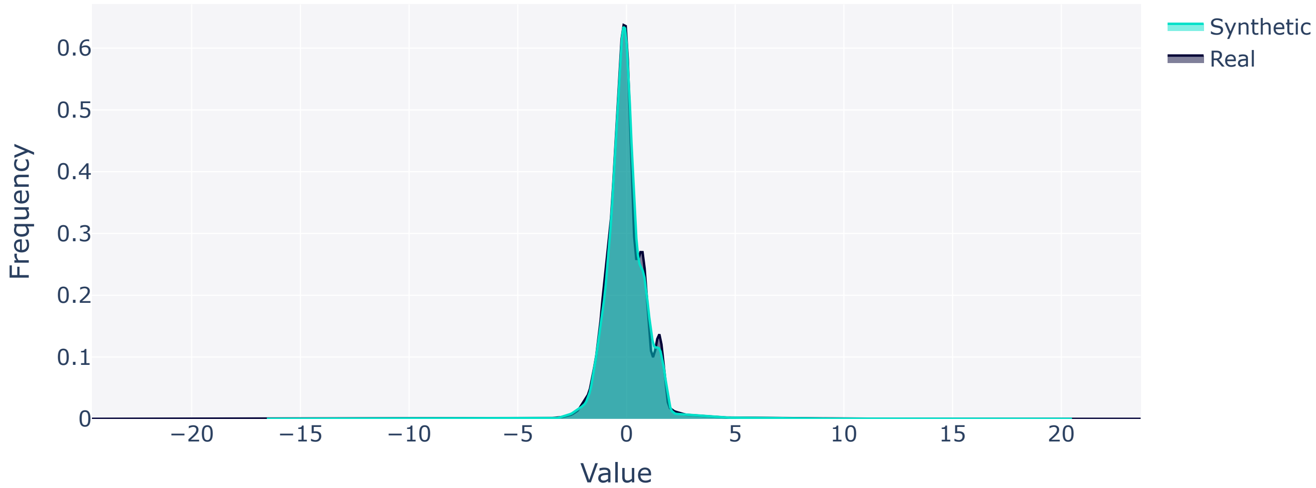
Real vs. Synthetic Data for column 'V8'



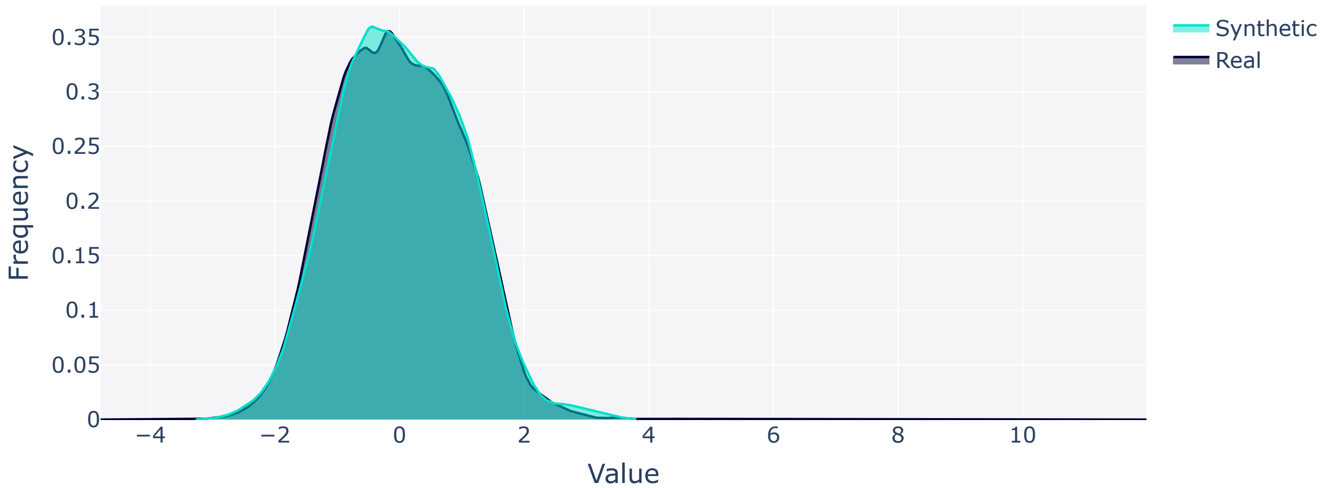
Real vs. Synthetic Data for column 'V9'



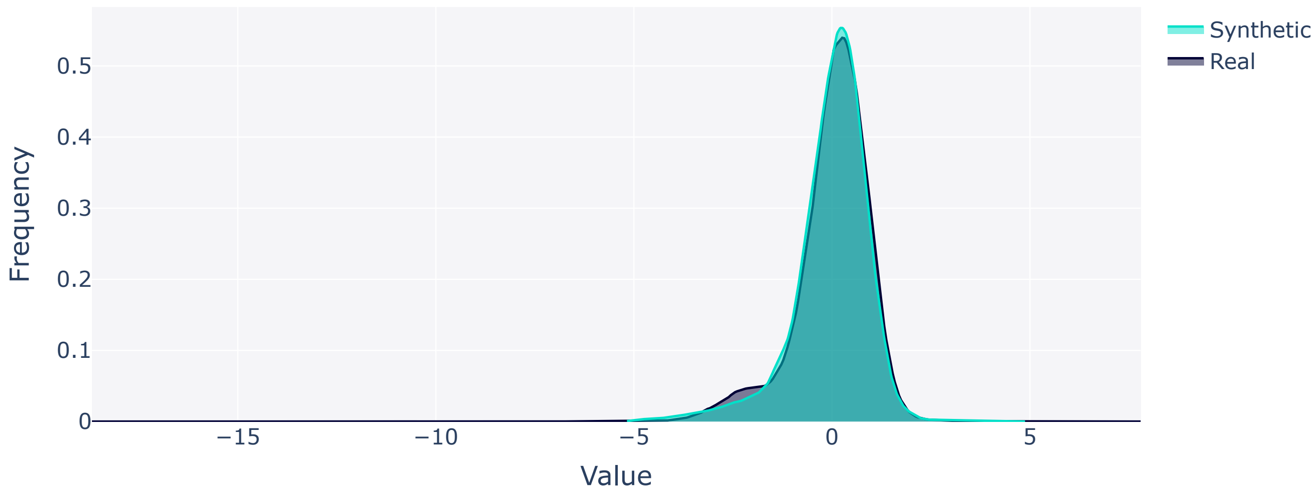
Real vs. Synthetic Data for column 'V10'



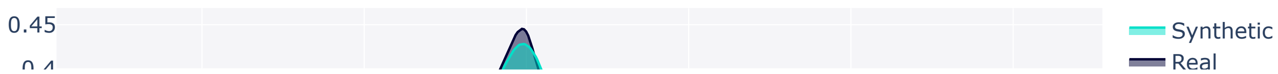
Real vs. Synthetic Data for column 'V11'

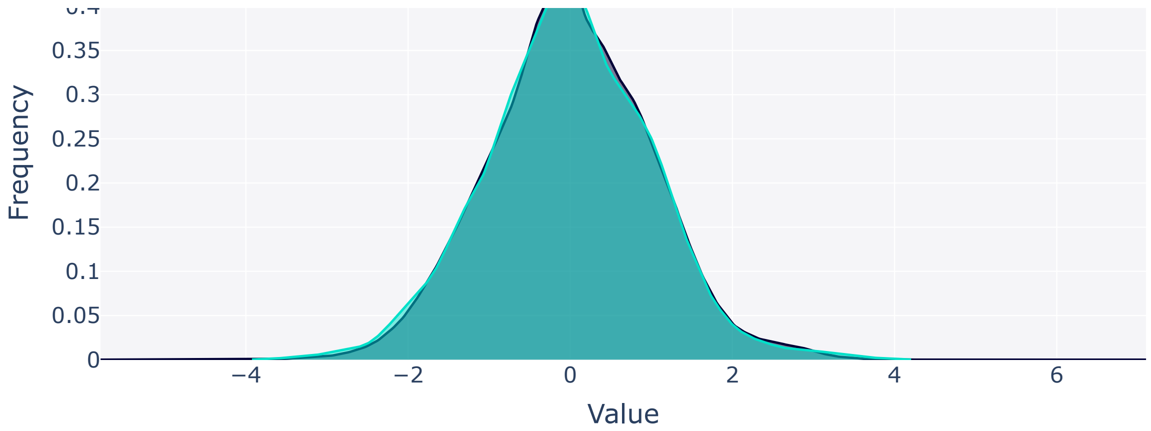


Real vs. Synthetic Data for column 'V12'

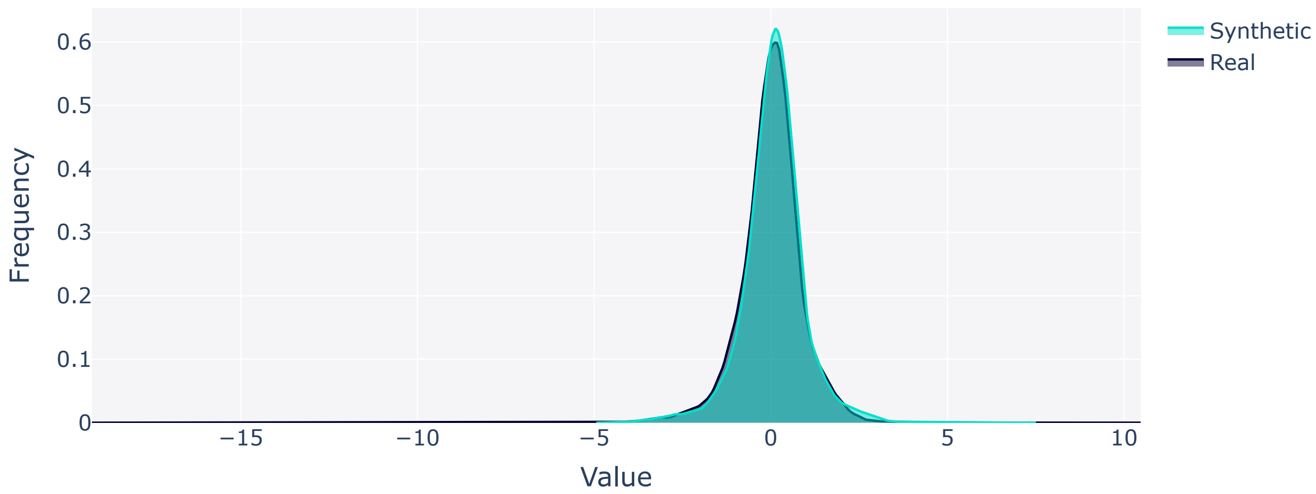


Real vs. Synthetic Data for column 'V13'

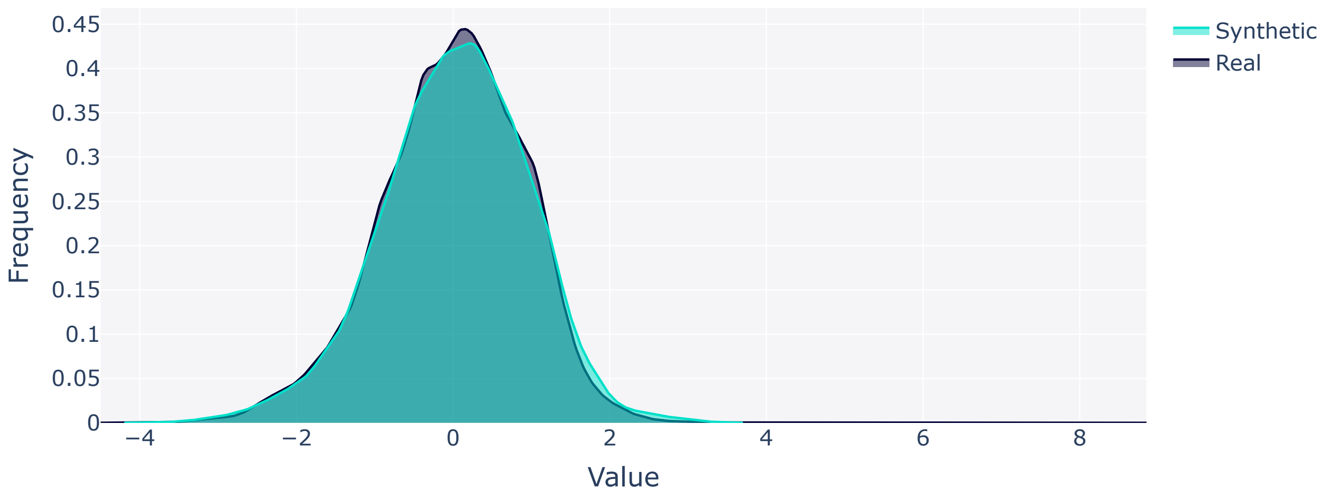




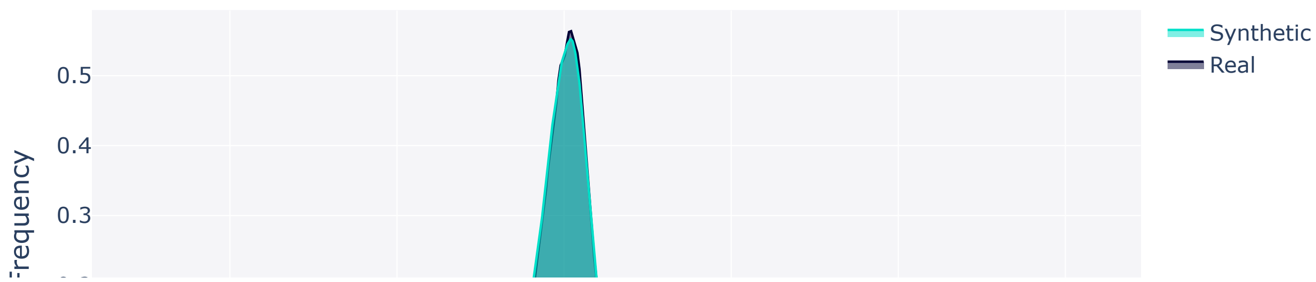
Real vs. Synthetic Data for column 'V14'

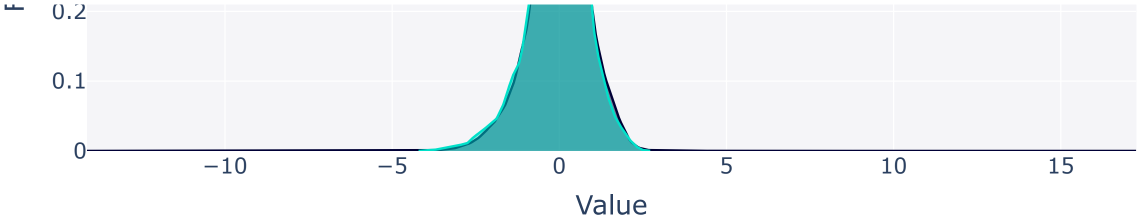


Real vs. Synthetic Data for column 'V15'

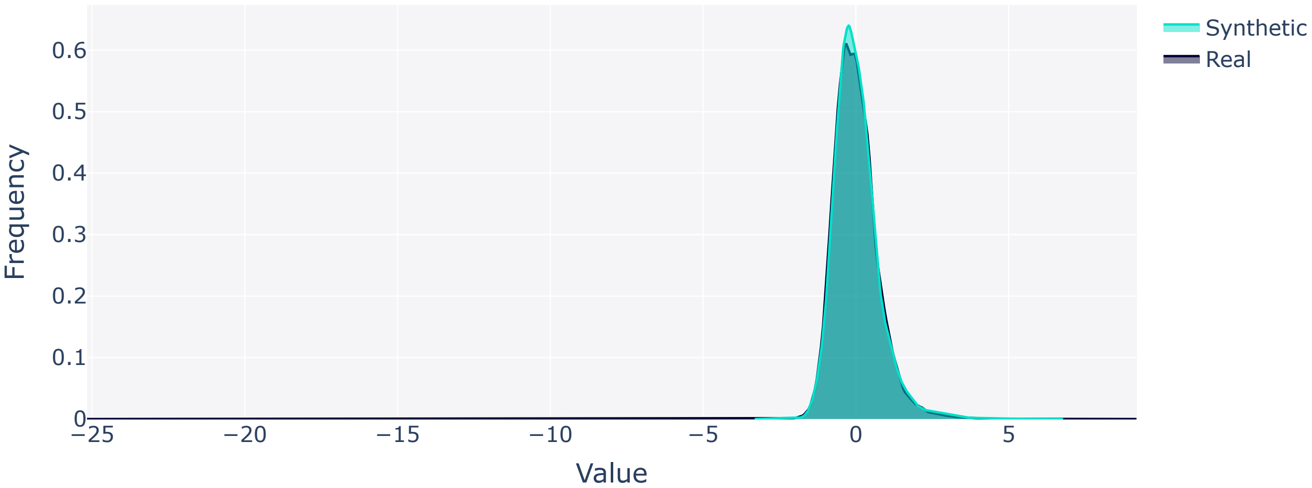


Real vs. Synthetic Data for column 'V16'

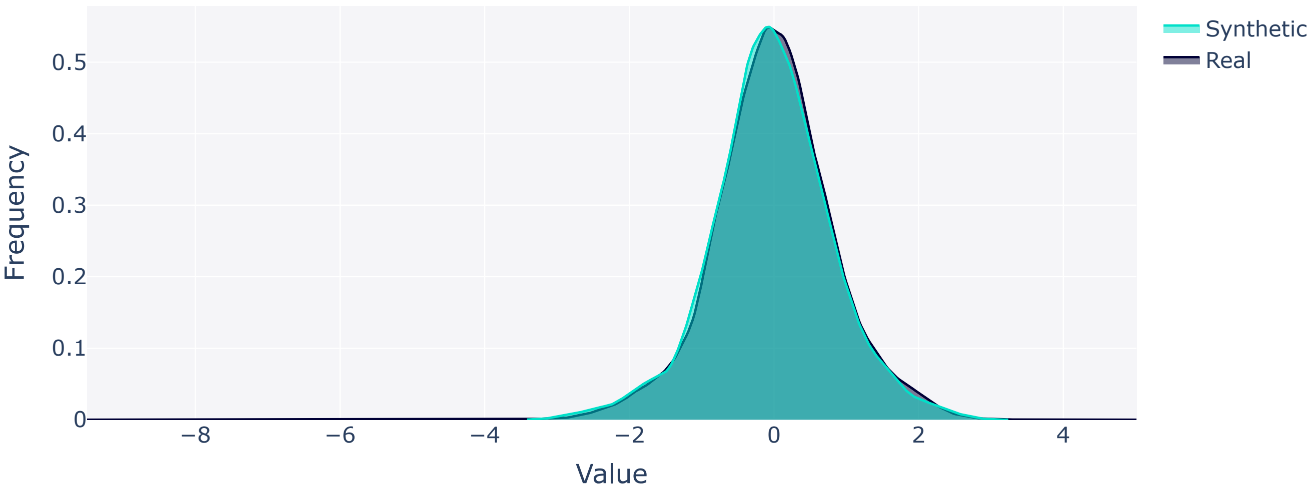




Real vs. Synthetic Data for column 'V17'

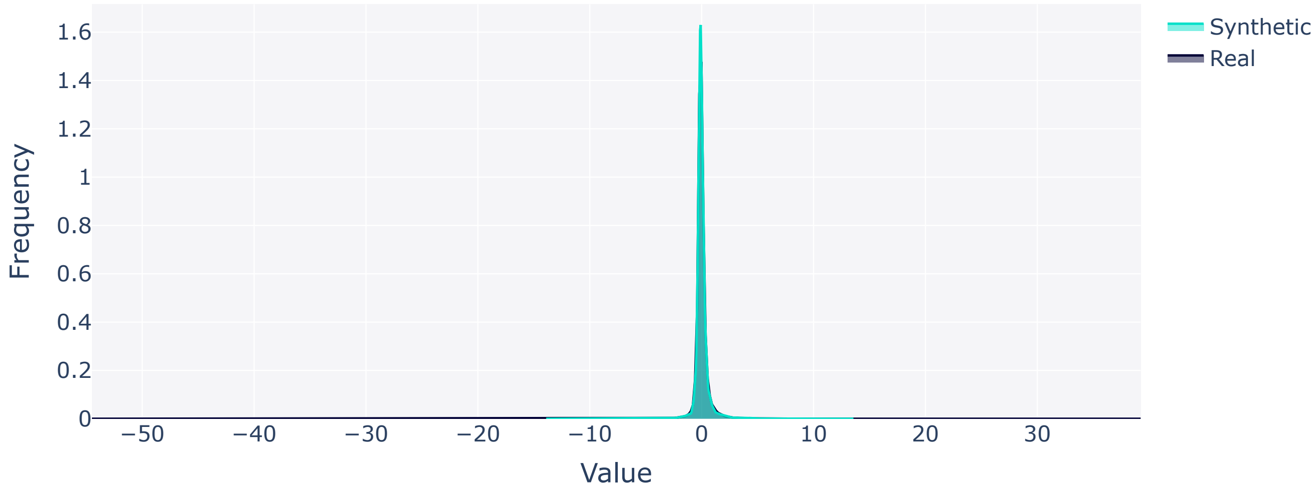


Real vs. Synthetic Data for column 'V18'

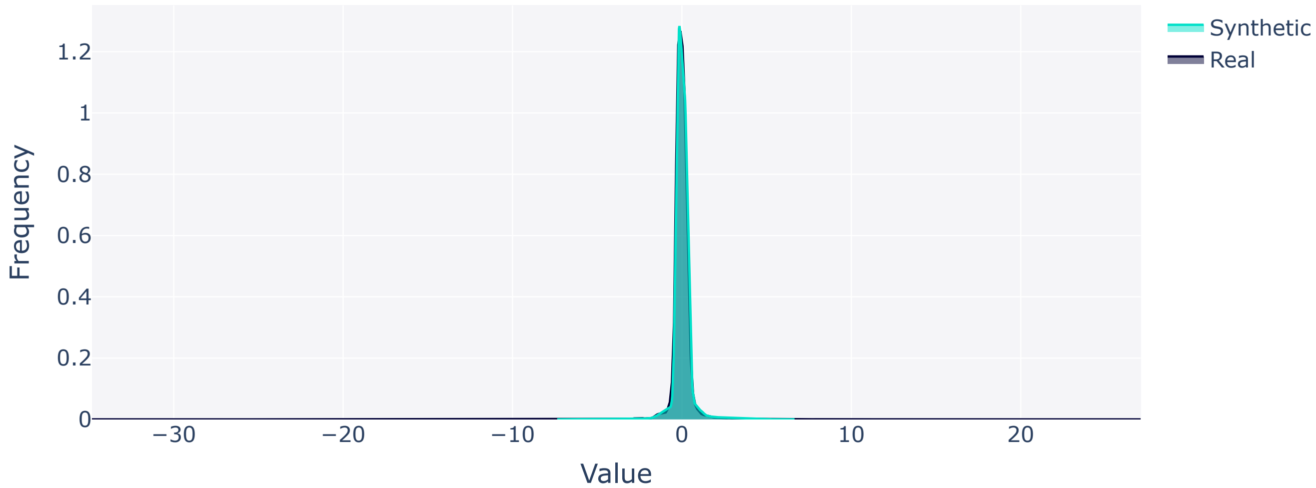


Real vs. Synthetic Data for column 'V19'

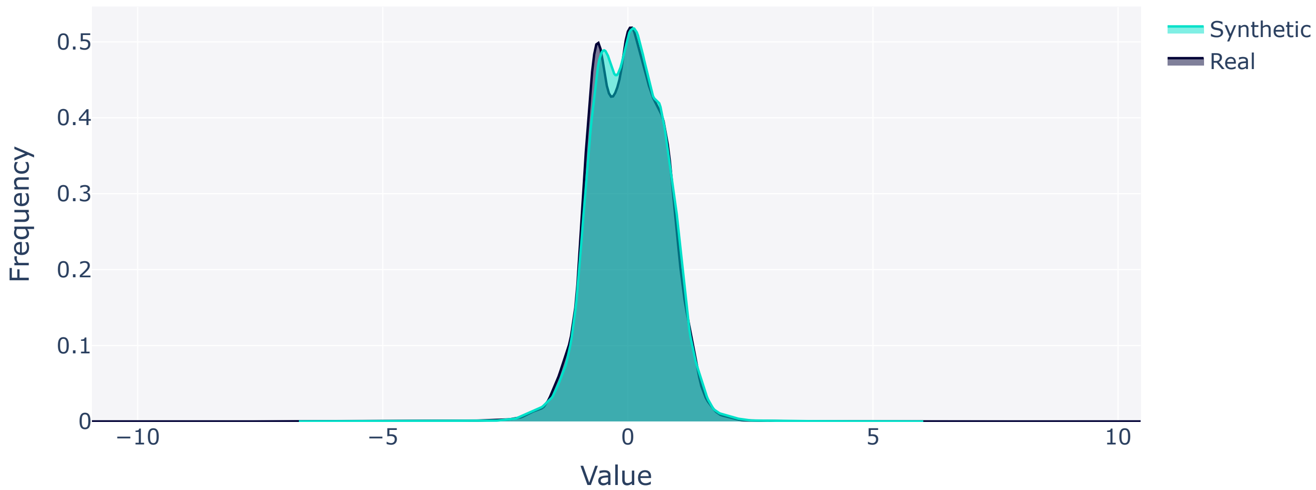
Real vs. Synthetic Data for column 'V20'



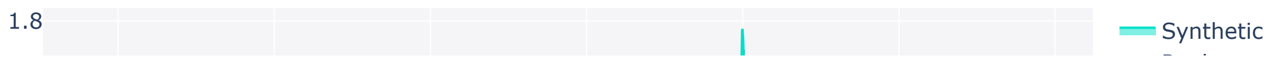
Real vs. Synthetic Data for column 'V21'

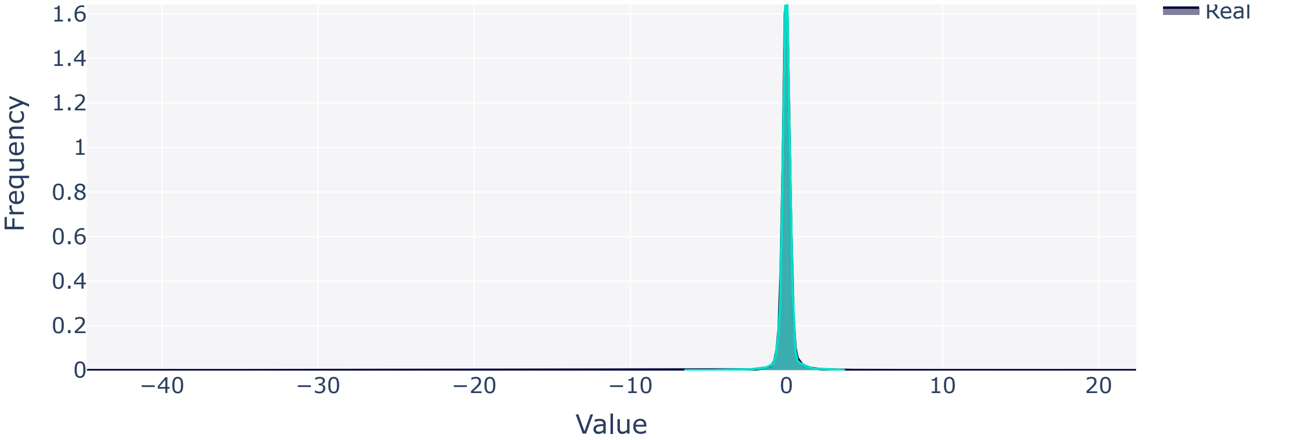


Real vs. Synthetic Data for column 'V22'

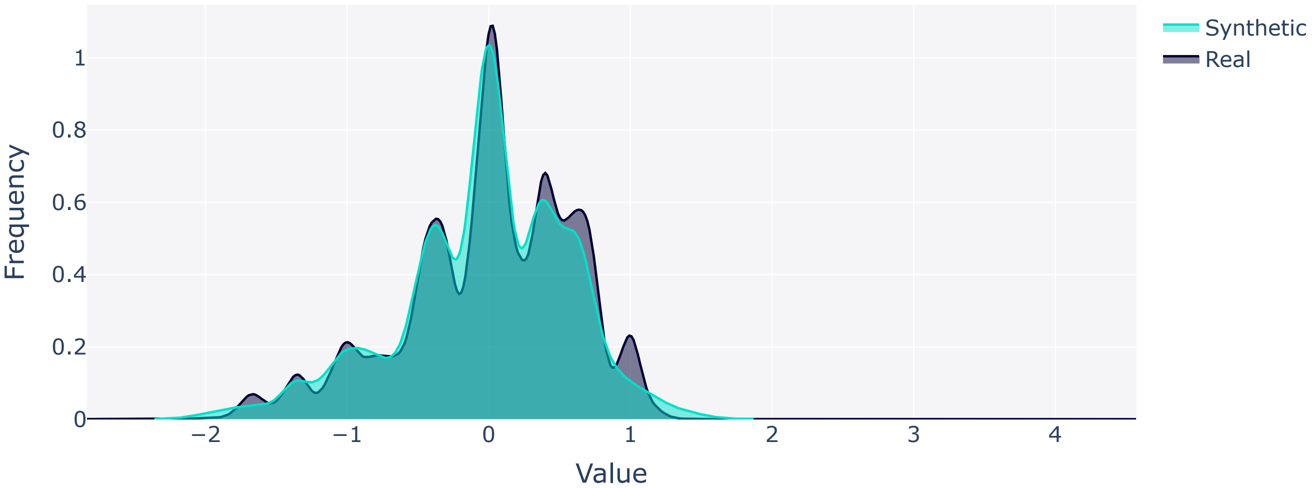


Real vs. Synthetic Data for column 'V23'

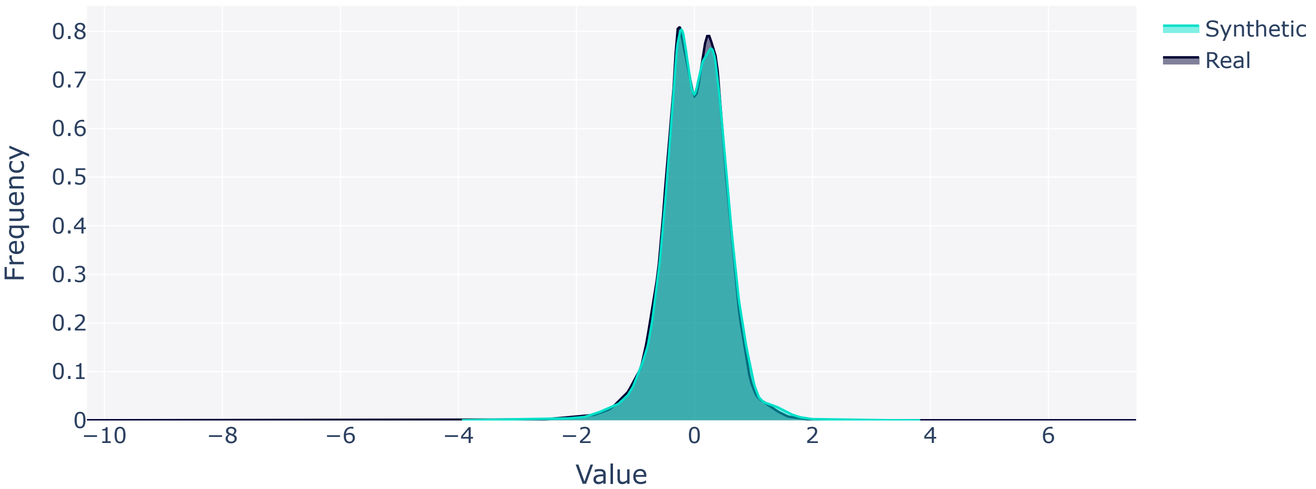




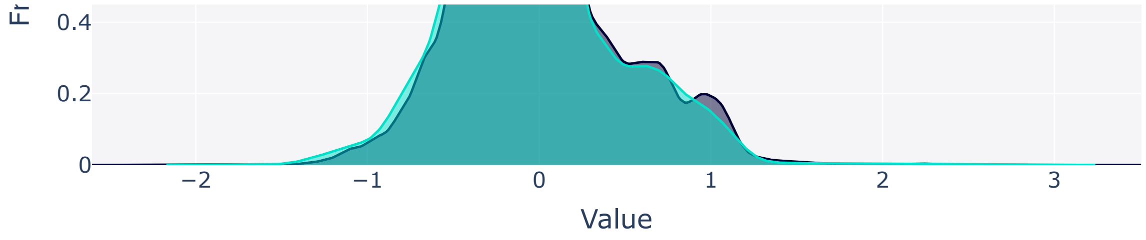
Real vs. Synthetic Data for column 'V24'



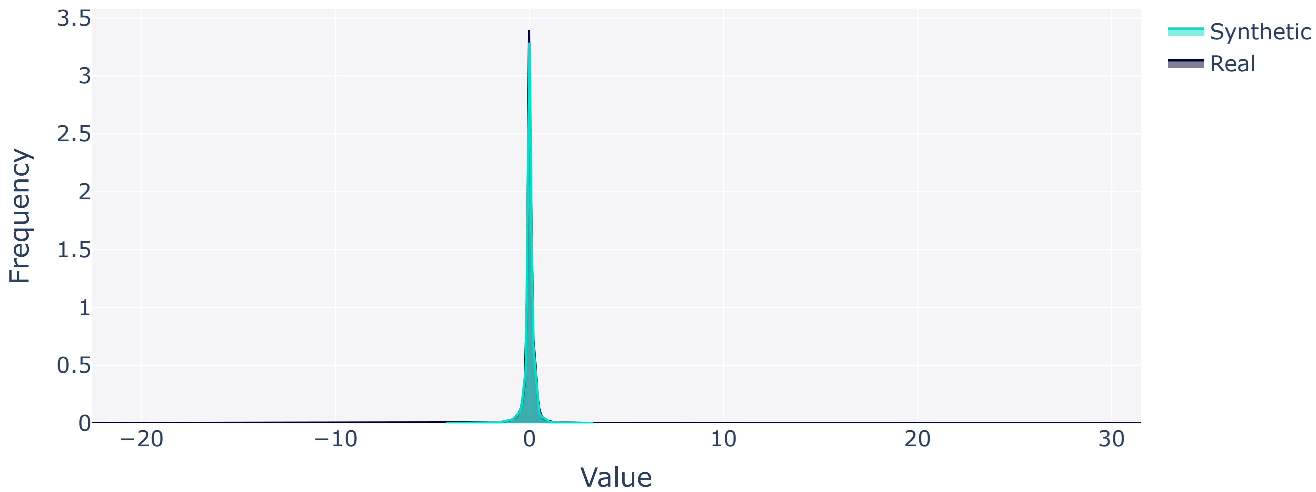
Real vs. Synthetic Data for column 'V25'



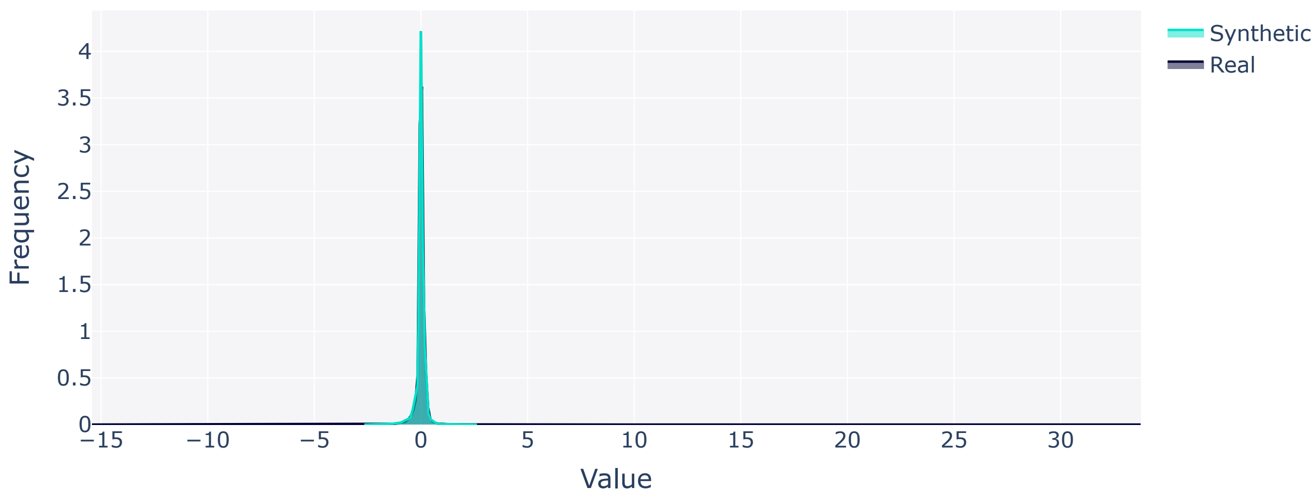
Real vs. Synthetic Data for column 'V26'



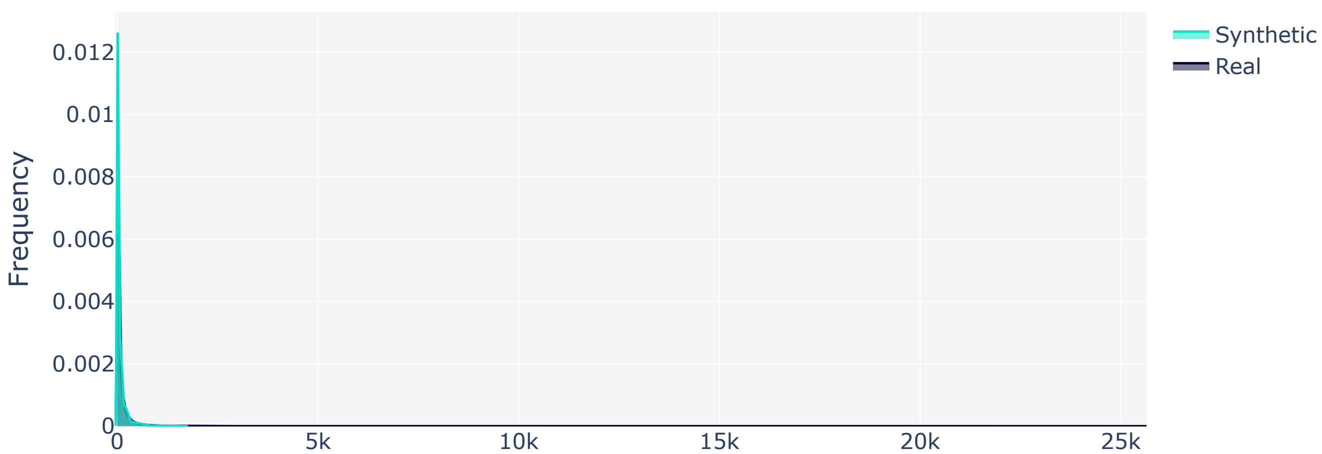
Real vs. Synthetic Data for column 'V27'



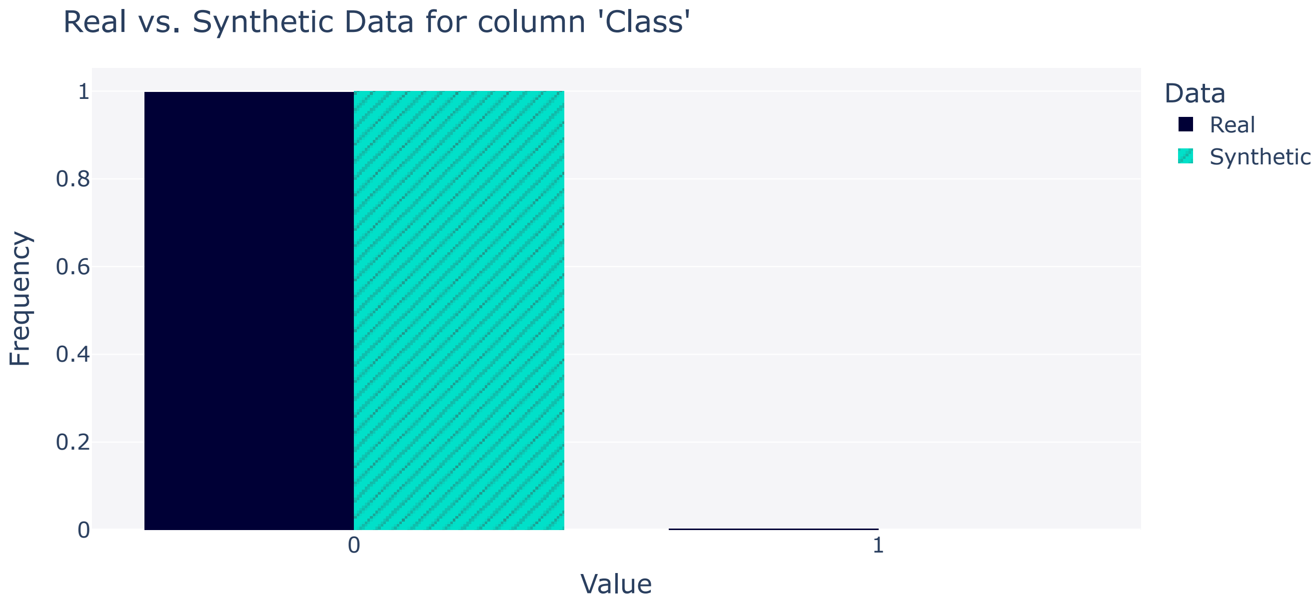
Real vs. Synthetic Data for column 'V28'



Real vs. Synthetic Data for column 'Amount'



Value



Real vs. Synthetic Data for column 'Class'
Frequency
1
0.8
0.6
0.4
0.2
0
0
1
Value
Data
Real
Synthetic

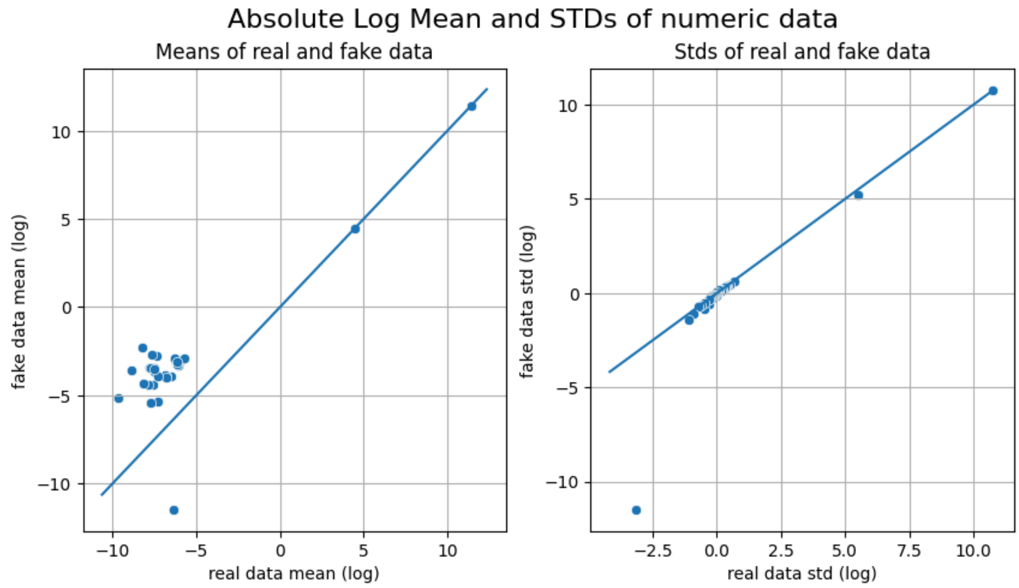
```
display(synthetic_data)
```

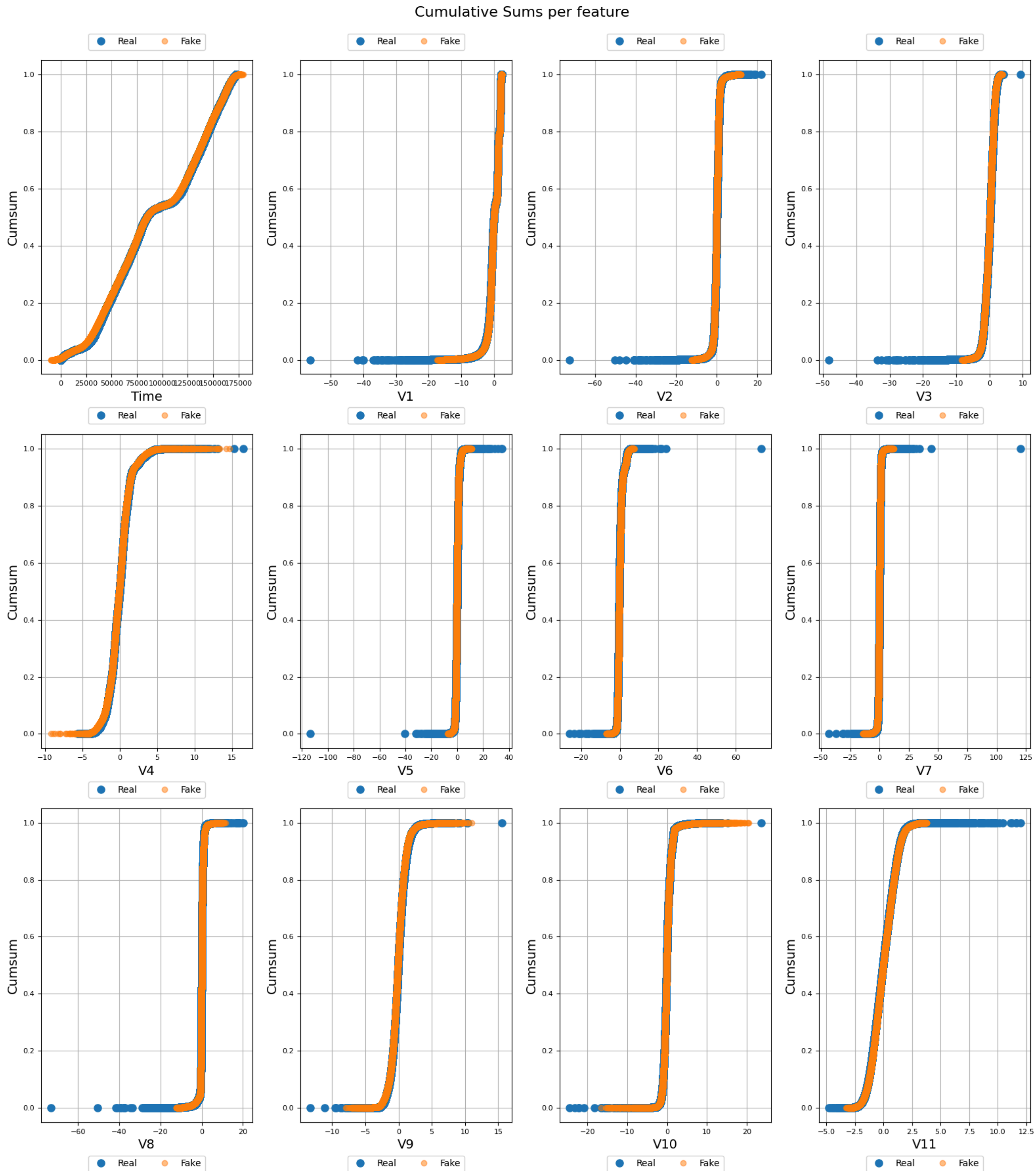
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23
0	91056.695925	-1.723675	0.852592	1.599076	0.977421	-0.634219	0.057729	0.389300	0.349830	-0.454323	...	0.029519	0.143773	-0.115416
1	120555.770719	0.945667	-1.694904	-0.643204	0.456008	-1.675624	0.235145	-0.794688	0.651628	0.761526	...	0.443907	0.203245	-0.225862
2	19211.431375	1.089114	-0.103770	-0.431534	0.777430	-0.160552	0.708622	-0.166989	1.038936	-0.766063	...	0.231437	0.403516	-0.121254
3	168567.683800	-1.639630	-0.271536	1.116587	-1.249007	-0.296331	0.019261	0.137442	0.520939	0.053335	...	0.446310	1.022763	0.194034
4	84286.992834	1.037571	-0.088699	0.630343	2.768153	-1.362145	-0.232675	-0.791321	0.199302	-0.120581	...	0.260395	0.103423	0.025870
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
199995	50834.196072	1.425029	-1.659942	-1.288374	-1.254835	-0.472430	-1.367811	-0.365032	-0.170695	-0.623026	...	0.091133	-0.405730	-0.057501
199996	168700.521584	2.012798	0.332498	-1.122220	1.470186	0.342357	-0.949444	0.326096	-0.309645	0.331347	...	-0.286460	-0.502850	0.349740
199997	11877.219890	1.331357	-1.351277	-0.153055	-0.929973	0.875562	4.016801	-0.981481	0.389933	0.379197	...	-0.017242	-0.286952	-0.216123
199998	8362.860422	-0.311491	0.889564	0.920301	-0.717985	0.464777	-0.700066	0.307352	0.006580	1.573938	...	0.295324	1.186391	-0.262274
199999	53088.791102	-0.301108	0.956909	1.415510	1.655014	0.846881	-1.179035	1.363445	-0.401781	-0.804008	...	0.177855	0.231027	-0.177608

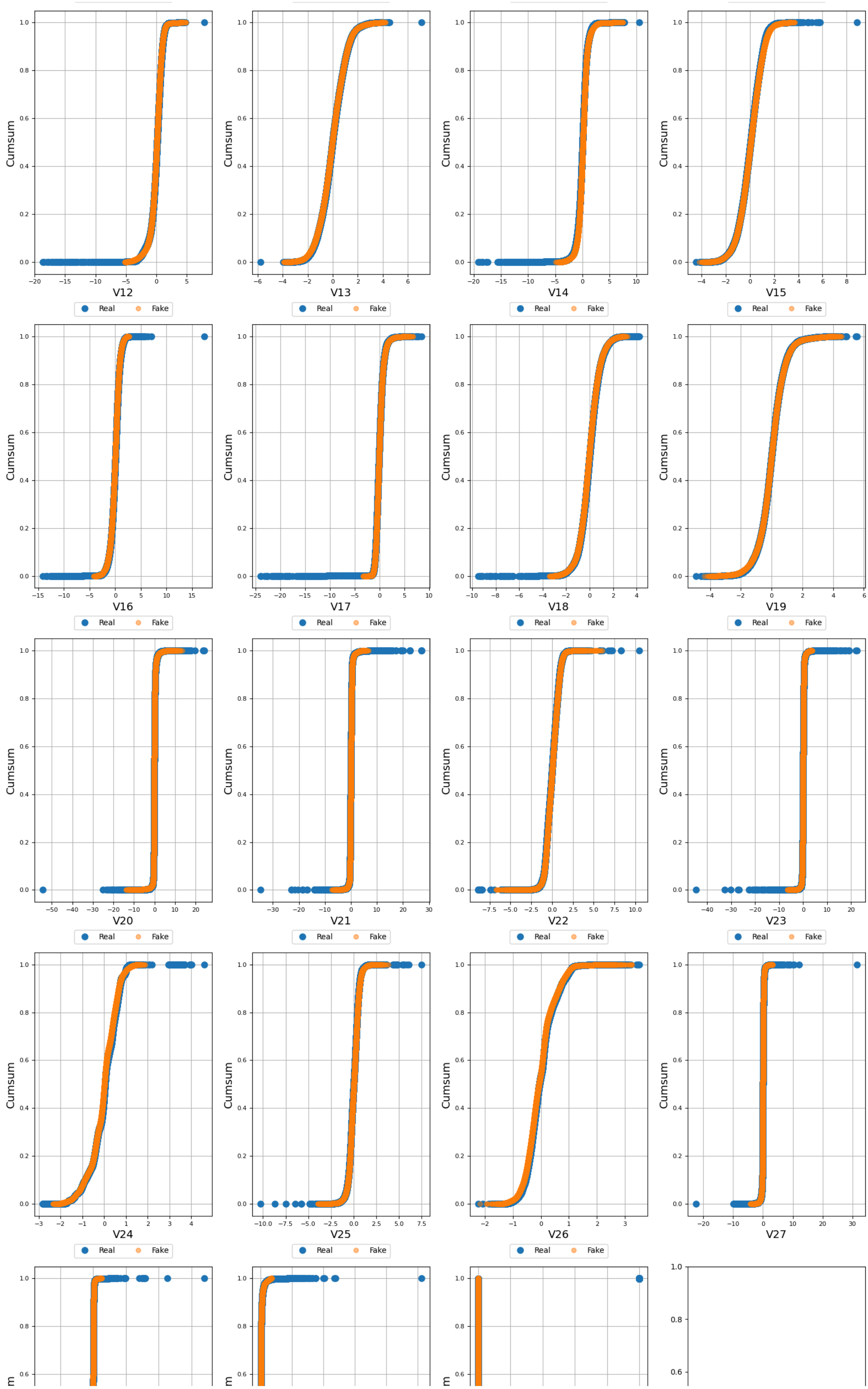
200000 rows × 31 columns

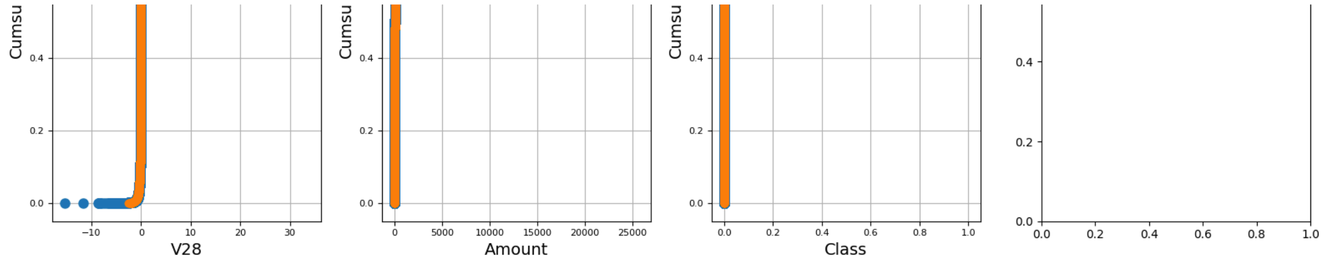
```
# from itertools import combinations
# from matplotlib.backends.backend_pdf import PdfPages

# # 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()
```

```
from table_evaluator import TableEvaluator

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

table_evaluator.visual_evaluation()
```

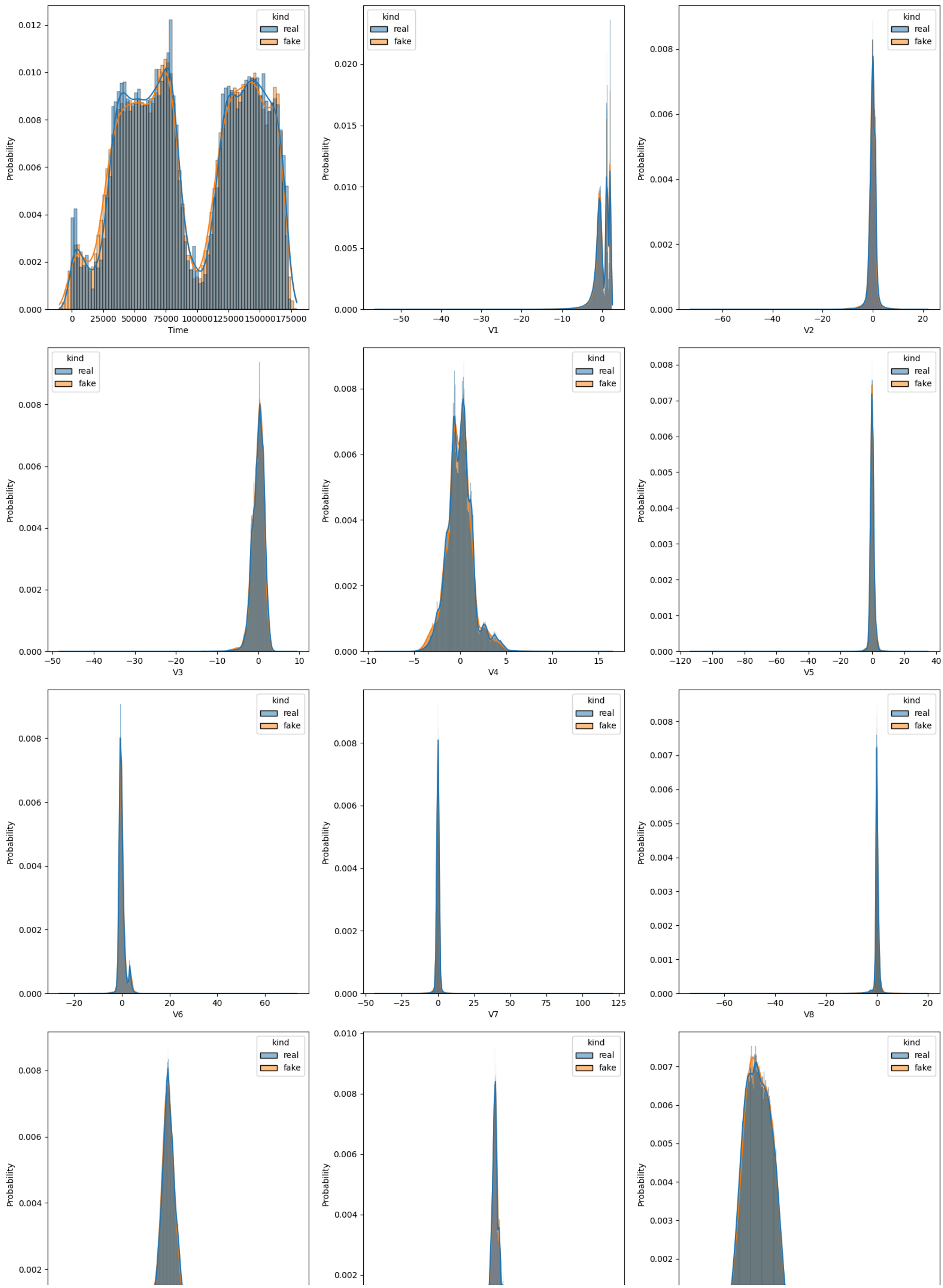

Absolute Log Mean and STDs of numeric data
Means of real and fake data
Stds of real and fake data
fake data mean (log)
real data mean (log)
fake data std (log)
real data std (log)

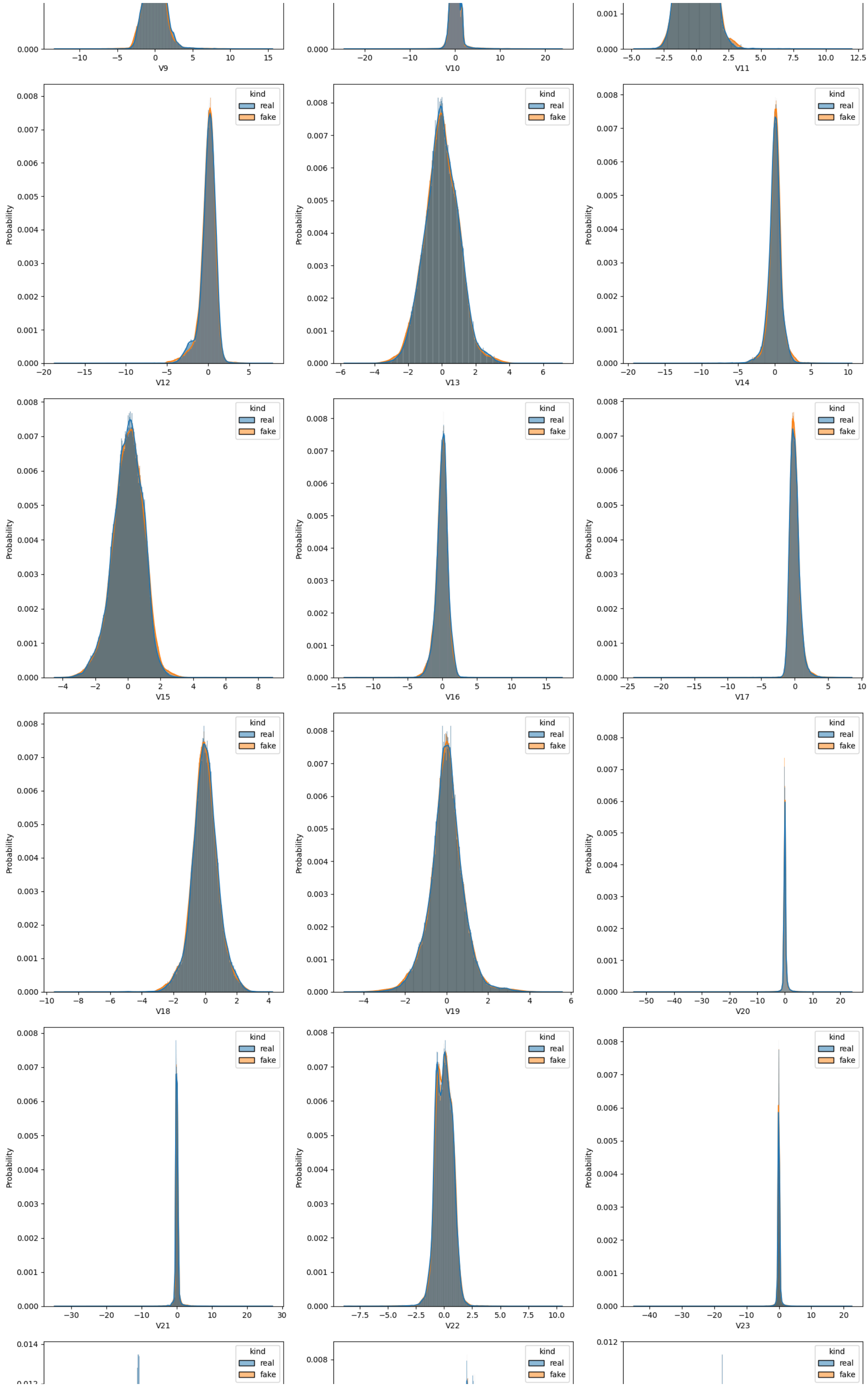

Cumulative Sums per feature
Real
Fake
Cumsum
Time
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11

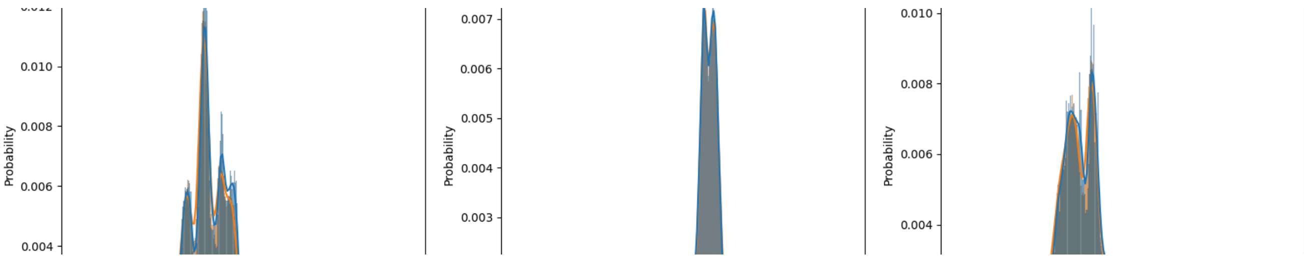

Cumsum
V12
V13
V14
V15
V16
V17
V18
V19
V20
V21
V22
V23
V24
V25
V26
V27
Real
Fake



Distribution per feature




V9
V10
V11
V12
V13
V14
V15
V16
V17
V18
V19
V20
V21
V22
V23
Probability
kind
real
fake


Probability
0.012
0.010
0.008
0.006
0.004
Probability
0.007
0.006
0.005
0.004
0.003
Probability
0.010
0.008
0.006
0.004