Jupyter Notebook for comparing synthetic data and metrics from the Synthetic Data Vault (SDV)

In [19]:

```python
import pandas as pd
import matplotlib.pyplot as plt
```

In [20]:

```python
# Load data sets for comparison
original = pd.read_csv("data/german_credit.csv")
synthetic = pd.read_csv("generated_data/synthethic2.csv")
```

In [21]:

```python
# Check sizes of both data sets
print("Size of the original data = {}".format(len(original)))
print("Size of the synthetic data = {}".format(len(synthetic)))
```

```
Size of the original data = 1000
Size of the synthetic data = 1000
```

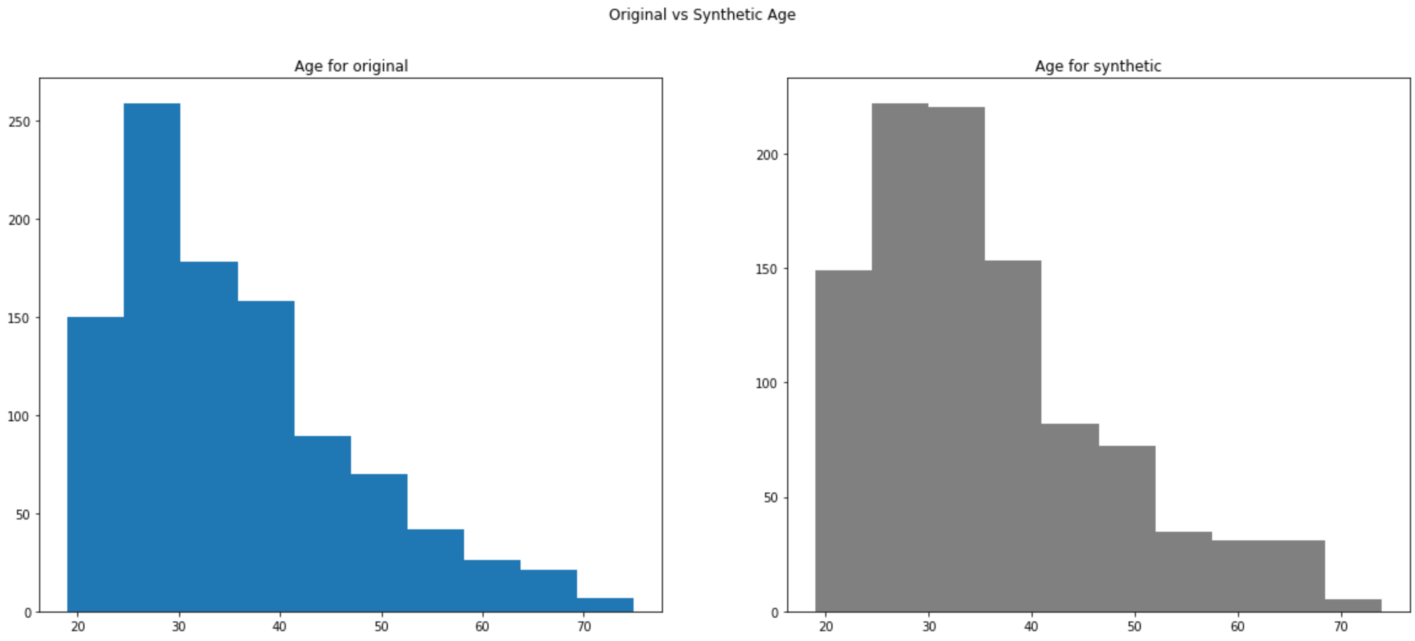
Visual comparison

In [22]:

```python
fig, (ax1, ax2) = plt.subplots(1,2, figsize=(20,8))
ax1.hist(original["Age..years."])
ax2.hist(synthetic["Age..years."], color="grey")
ax1.set_title("Age for original")
ax2.set_title("Age for synthetic")
fig.suptitle('Original vs Synthetic Age')
```

Out[22]:

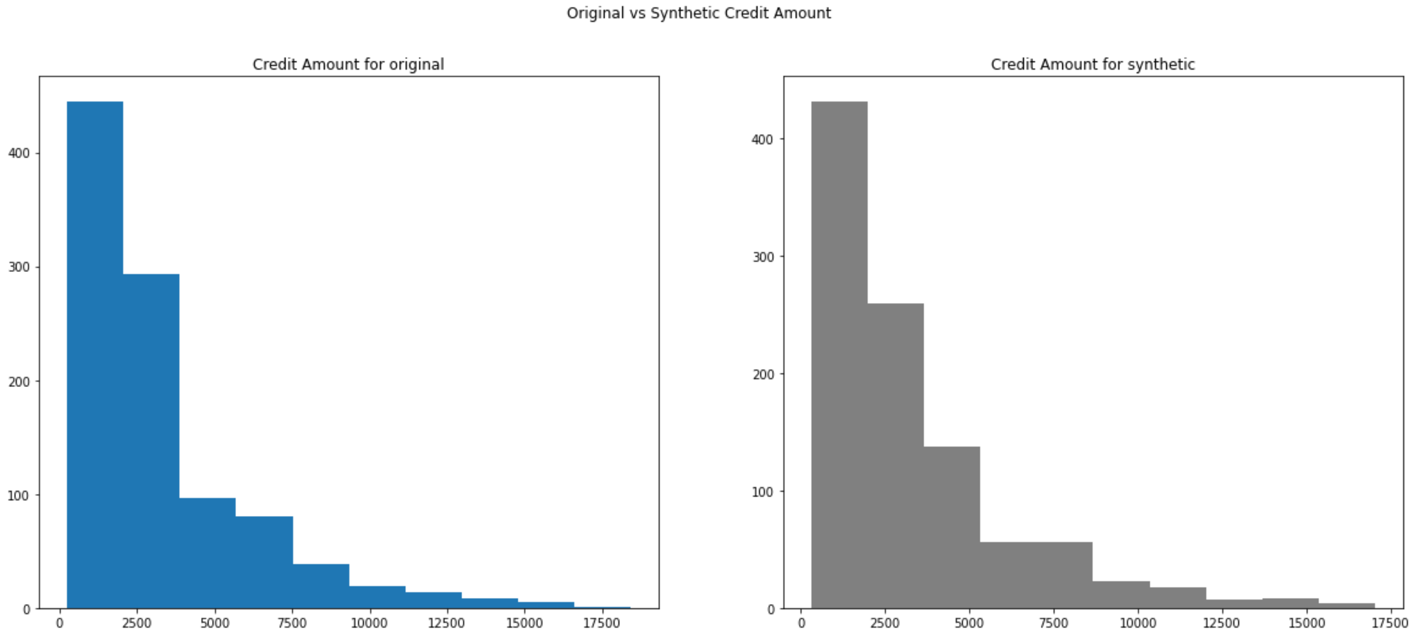
```
Text(0.5, 0.98, 'Original vs Synthetic Age')
```



In [23]:

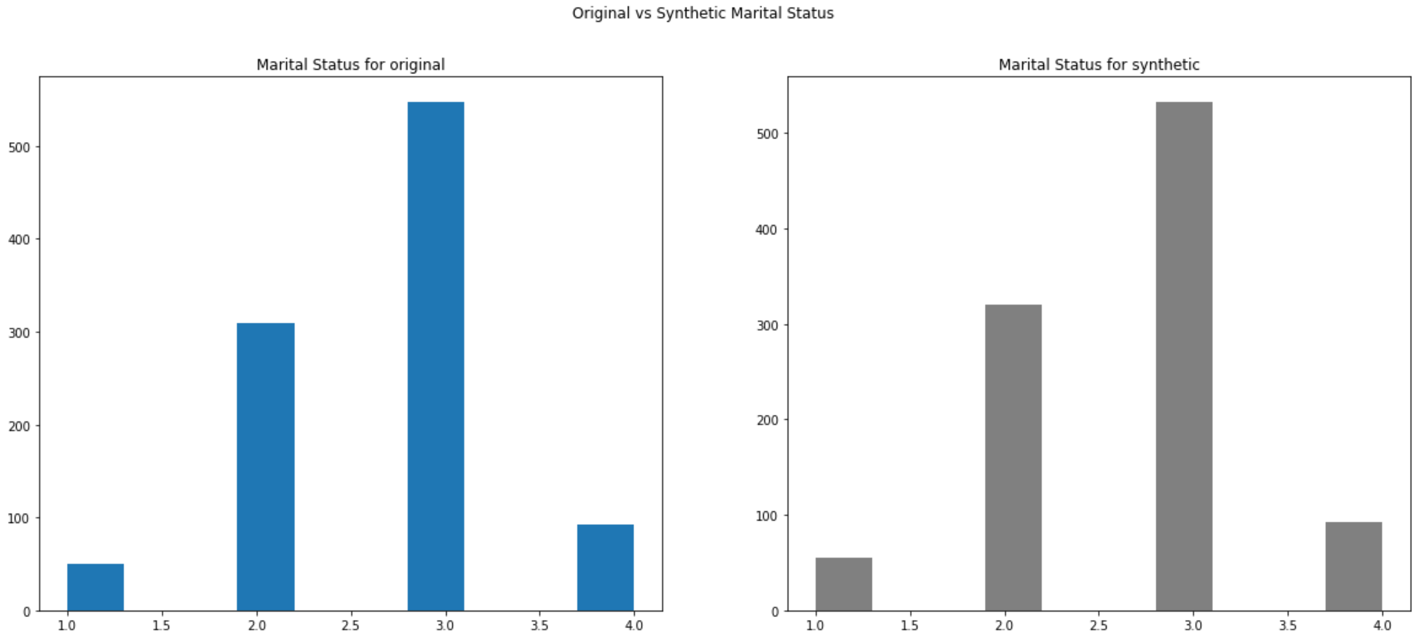
```python
fig, (ax1, ax2) = plt.subplots(1,2, figsize=(20,8))
ax1.hist(original["Credit.Amount"])
ax2.hist(synthetic["Credit.Amount"], color="grey")
fig.suptitle('Original vs Synthetic Credit Amount')
ax1.set_title("Credit Amount for original")
ax2.set_title("Credit Amount for synthetic")
```

```
Out[23]: Text(0.5, 1.0, 'Credit Amount for synthetic')
```



```
In [24]: fig, (ax1, ax2) = plt.subplots(1,2, figsize=(20,8))
         ax1.hist(original["Sex...Marital.Status"])
         ax2.hist(synthetic["Sex...Marital.Status"], color="grey")
         fig.suptitle('Original vs Synthetic Marital Status')
         ax1.set_title("Marital Status for original")
         ax2.set_title("Marital Status for synthetic")
```

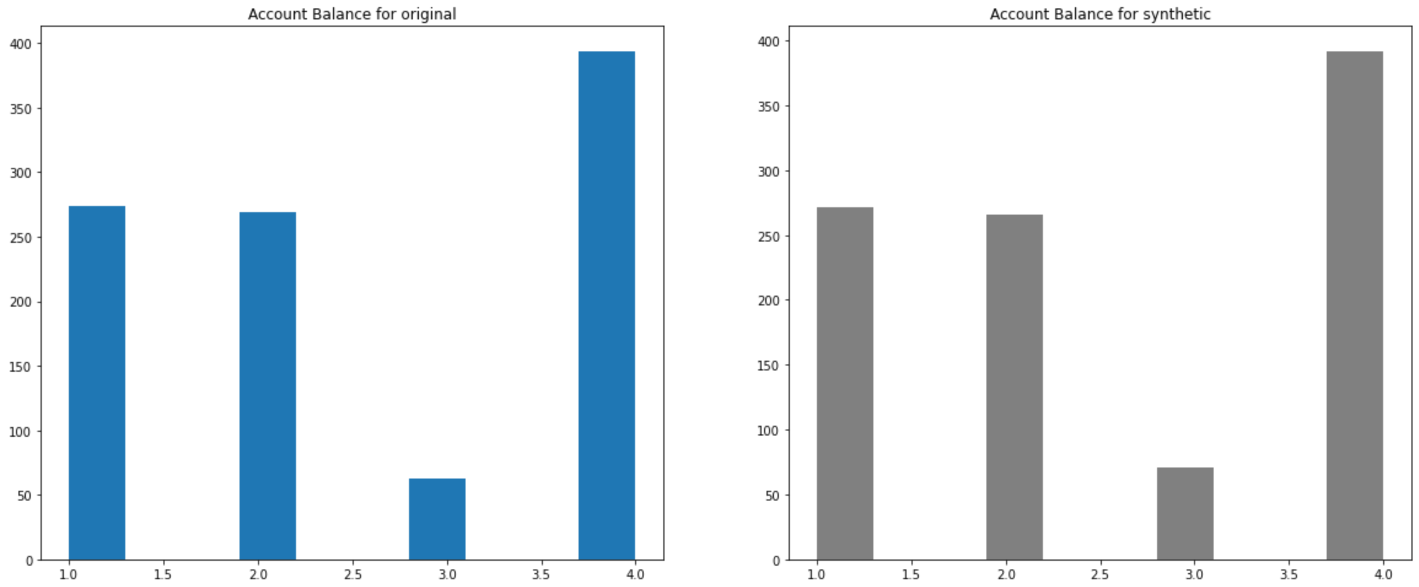
```
Out[24]: Text(0.5, 1.0, 'Marital Status for synthetic')
```



```
In [25]: fig, (ax1, ax2) = plt.subplots(1,2, figsize=(20,8))
         ax1.hist(original["Account.Balance"])
         ax2.hist(synthetic["Account.Balance"], color="grey")
         fig.suptitle('Original vs Synthetic Account Balance')
         ax1.set_title("Account Balance for original")
         ax2.set_title("Account Balance for synthetic")
```

```
Out[25]: Text(0.5, 1.0, 'Account Balance for synthetic')
```

Original vs Synthetic Account Balance



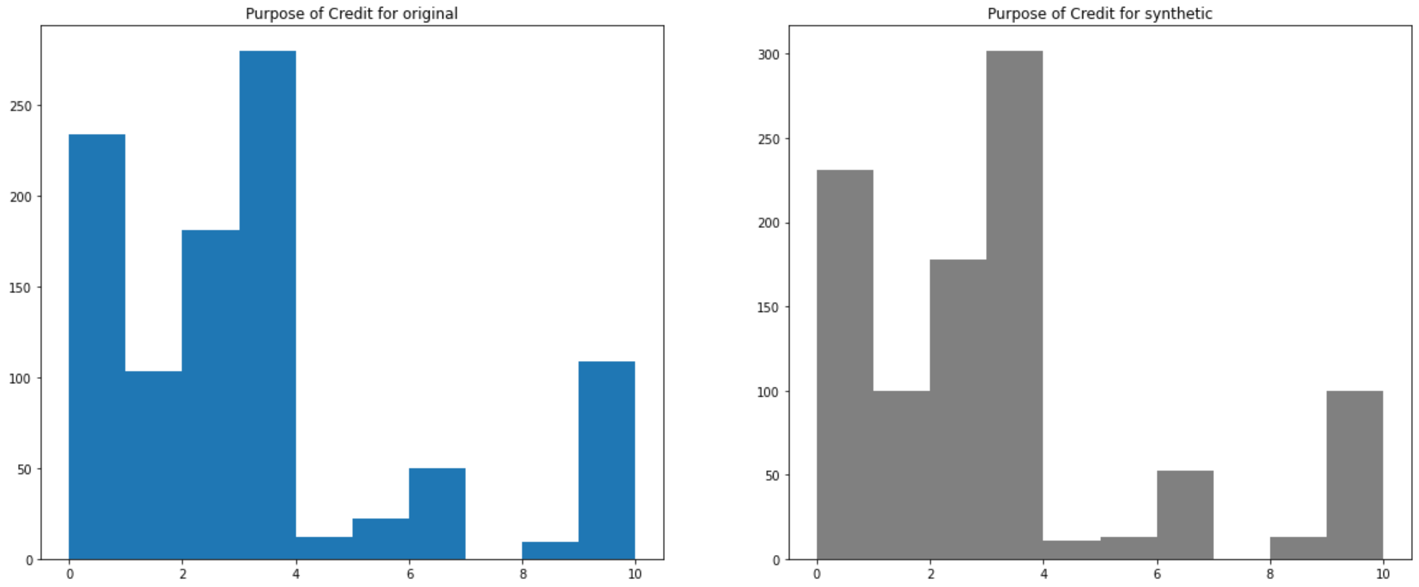
In [26]:

```
fig, (ax1, ax2) = plt.subplots(1,2, figsize=(20,8))
ax1.hist(original["Purpose"])
ax2.hist(synthetic["Purpose"], color="grey")
fig.suptitle('Original vs Synthetic Purpose of Credit')
ax1.set_title("Purpose of Credit for original")
ax2.set_title("Purpose of Credit for synthetic")
```

Out[26]:

```
Text(0.5, 1.0, 'Purpose of Credit for synthetic')
```

Original vs Synthetic Purpose of Credit



Metrics for Synthetic Data Vault (SDV)

In [27]:

```
from sdv.metrics.tabular import MulticlassDecisionTreeClassifier, LinearRegression, Binary
from sdv.evaluation import evaluate
from sdv.metrics.tabular import CSTest, KSTest

#
```

How well the data does when it comes to Machine Learning Models

In [28]:

```
BinaryDecisionTreeClassifier.compute(original, synthetic, target='Creditability')
```

Out[28]:

```
0.7657466383581034
```

In [29]:

```
MulticlassDecisionTreeClassifier.compute(original, synthetic, target='Creditability')
```

Out[29]:

```
0.6031930447962879
```

How well the original data does when it comes to Machine Learning Models

In [30]:

```
# 70:30 cross validation on the real data-set example
train = original.sample(int(len(original) * 0.75))

test = original[~original.index.isin(train.index)]
```

In [31]:

```
MulticlassDecisionTreeClassifier.compute(test, train, target='Creditability')
```

Out[31]:

```
0.6256910319410319
```

In [32]:

```
BinaryDecisionTreeClassifier.compute(test, train, target='Creditability')
```

Out[32]:

```
0.7780979827089337
```

Statistical metric

In [33]:

```
# https://sdv.dev/SDV/user_guides/evaluation/single_table_metrics.html
KSTest.compute(original, synthetic)
```

Out[33]:

```
0.9854285714285714
```