Step 1: Import libraries

```
# Import necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats

# Ensure plots are displayed inline in the notebook
%matplotlib inline

print("Libraries imported successfully!")
```

```
Libraries imported successfully!
```

Step 2: Download the dataset

Please note that the dataset that will be downloaded below is a dummy dataset that has been designed for the tutorial. You need to use the actualy dataset provided to you for the analysis

```
!gdown 1f9ewHqDTGo45XIVH16-benkMWsAHBAZr
```

```
Downloading...
From: https://drive.google.com/uc?id=1f9ewHqDTGo45XIVH16-benkMWsAHBAZr
To: /content/Copy of compiled_risk_data.xlsx
100% 310k/310k [00:00<00:00, 106MB/s]
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
print("Setup complete. Imported pandas, seaborn, and matplotlib. Downloaded compile risk dataset.")
```

```
Setup complete. Imported pandas, seaborn, and matplotlib. Downloaded compile risk dataset.
```

Step 3: Load the Data Section

Now even though we have downloaded the dataset, we still need to load it into our Python environment. For this we will utilize the Pandas library.

```
# Loading the dataset

df = pd.read_excel('Copy of compiled_risk_data.xlsx')

# Display the first five rows of the dataframe
print(df.head())
```

```
     project_name                     Smart contract address  \
0  Data Analytics    384571416209d08623c6ace9422613fc8970475d
1  Data Analytics   0xAb5801a7D398351b8bE11C439e05C5B3259aeB9B
2  Data Analytics  0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db
3  Data Analytics  0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB
4  Data Analytics  0x617F2E2fD72FD9D5503197092aC168c91465E7f2

                                      Blog post link  \
0  https://chainsecurity.com/security-audit/circl...
1  https://stackoverflow.com/questions/75030483/w...
2  https://stackoverflow.com/questions/71115106/s...
3  https://stackoverflow.com/questions/75030483/w...
4  https://stackoverflow.com/questions/69466137/h...

                                       Audit website     Chain  \
0                          https://chainsecurity.com  Ethereum
1  https://studygroup.moralis.io/t/compilation-er...  Ethereum
2  https://ethereum.stackexchange.com/questions/1...  Ethereum
3  https://studygroup.moralis.io/t/compilation-er...  Ethereum
4  https://ethereum.stackexchange.com/questions/1...  Ethereum

  Is_closed_source  hidden_owner  anti_whale_modifiable  Is_anti_whale  \
0            False         False                  False          False
1            False         False                   True           True
2             True         False                  False           True
```

```
3             True         False                  False          False
4             True         False                   True          False

   Is_honeypot  ...  centralized_risk_high  centralized_risk_low  \
0        False  ...                  False                 False
1        False  ...                  False                  True
2        False  ...                  False                  True
3         True  ...                  False                  True
4         True  ...                  False                 False

   event_setter  external_dependencies  immutable_states  \
0          True                   True              True
1         False                   True             False
2         False                   True             False
3         False                  False              True
4          True                   True             False

   reentrancy_without_eth_transfer  incorrect_inheritance_order  \
0                             True                        False
1                            False                         True
2                            False                        False
3                             True                        False
4                             True                        False

   shadowing_local  events_maths  \
0            False         False
1            False          True
2             True          True
3            False          True
4            False          True

          Summary/rationale of risk tags marked true
0  Bad Contract: Assigned for flaws that indicate...
```

Calculate Correlation

To calculate the Phi coefficient, which is suitable for pairs of binary variables, we first need to establish a function that can handle this calculation:

```
def phi_coefficient(x, y):
    """Calculate the Phi coefficient for two binary variables."""
    # Create a contingency table
    contingency_table = pd.crosstab(x, y)
    # Calculate the phi coefficient from the contingency table
    chi2 = scipy.stats.chi2_contingency(contingency_table, correction=False)[0]
    n = np.sum(np.sum(contingency_table))
    phi = np.sqrt(chi2 / n)
    return phi

# Example calculation between two risk tags
phi = phi_coefficient(df['Is_honeypot'], df['anti_whale_modifiable'])
print(f"Phi Coefficient between 'Is_honeypot' and 'anti_whale_modifiable': {phi}")
```

```
Phi Coefficient between 'Is_honeypot' and 'anti_whale_modifiable': 0.43014356785902874
```

Phi value close to 0 indicates no correlation between the two columns.

Note: Phi values range from -1 to 1. A negative value of Phi indicates that the variables are inversely related, or when one variable increases, the other decreases. On the other hand, positive values indicate that when one variable increases, so does the other.

Let's now define the risk columns of our dataset.

```
risk_columns = ['Is_closed_source', 'hidden_owner', 'anti_whale_modifiable',
       'Is_anti_whale', 'Is_honeypot', 'buy_tax', 'sell_tax',
       'slippage_modifiable', 'Is_blacklisted', 'can_take_back_ownership',
       'owner_change_balance', 'is_airdrop_scam', 'selfdestruct', 'trust_list',
       'is_whitelisted', 'is_fake_token', 'illegal_unicode', 'exploitation',
       'bad_contract', 'reusing_state_variable', 'encode_packed_collision',
       'encode_packed_parameters', 'centralized_risk_medium',
       'centralized_risk_high', 'centralized_risk_low', 'event_setter',
       'external_dependencies', 'immutable_states',
       'reentrancy_without_eth_transfer', 'incorrect_inheritance_order',
       'shadowing_local', 'events_maths']
```

Now we will calculate the phi coefficient for all the columns

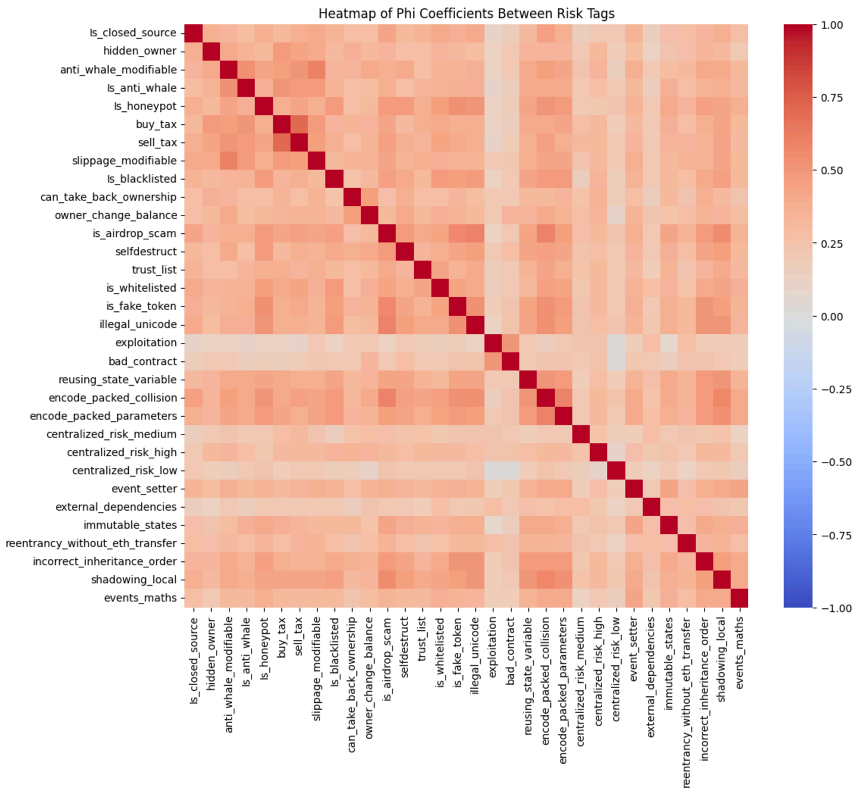
```
risk_df = df[risk_columns]

# Create a DataFrame to store Phi coefficients
phi_matrix = pd.DataFrame(index=risk_df.columns, columns=risk_df.columns)

# Calculate Phi coefficient for each pair of binary variables
for var1 in risk_df.columns:
    for var2 in risk_df.columns:
        phi_matrix.loc[var1, var2] = phi_coefficient(risk_df[var1], risk_df[var2])

print("Phi coefficients calculated for all pairs of variables:")
phi_matrix
```

Now even though we have the full correlation matrix in front of us, it is very difficult to visualize. One thing that we can do is only display those correlations where value is significantly positive or negative.

But a much better way is to visualize this matrix as a heatmap.

```
# Setting the size of the plot
plt.figure(figsize=(12, 10))
```

```
# Creating a heatmap
sns.heatmap(phi_matrix.astype(float), annot=False, fmt=".2f", cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Heatmap of Phi Coefficients Between Risk Tags')
plt.show()
```



You can experiment with a variety of versions of this heatmap to improve visibility of the trends

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```