4.12

Analyze the weighted network dataset available in this book's GitHub repository to study the relationship between degree and strength.

For undirected networks, measure the Pearson correlation coefficient between the degree and strength of all nodes.

For directed networks, do the same for in/out degree and in/out strength.

Do nodes with a high number of heights also have large strengths?

(ex. use python 3.11.9)

In [12]:
```
import pandas as pd
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
import gzip
```

In [13]:
```
# List of datasets with attributes
datasets = pd.DataFrame([
    ('Facebook Northwestern University', '', './socfb-Northwestern25/socfb-North
    ('IMDB movies and actors', '', './imdb/actors_movies.edges.gz'),
    ('IMDB actors costar', 'W', './imdb/actors_costar.edges.gz'),
    ('Twitter US politics', 'DW', './icwsm_polarization/retweet-digraph.edges.gz
    ('Enron Email', 'DW', './email-Enron/email-Enron.edges.gz'),
    ('Enron Executive Email', '', './ia-enron-only/ia-enron-only.edges'),
    ('Wikipedia math', 'D', './enwiki_math/enwiki_math.edges.gz'),
    ('Internet routers', '', './tech-RL-caida/tech-RL-caida.edges.gz'),
    ('US air transportation', '', './openflights/openflights_usa.edges.gz'),
    ('World air transportation', '', './openflights/openflights_world.edges.gz')
    ('Yeast protein interactions', '', './bio-yeast-protein-inter/bio-yeast-prot
    ('C. elegans brain', 'DW', './celegansneural/celegansneural.edges'),
    ('Everglades ecological food web', 'DW', './eco-everglades/eco-everglades.ed
], columns=['Name', 'Type', 'File'])
```

In [14]:
```
df = datasets.set_index('Name')
```

Degree of a Node:

> The degree of a node is the number of edges (or connections) it has in the network. For example, in a social network, a node (person) with a high degree would have many friends or connections.

Strength of a Node:

> The strength of a node refers to the total weight of the edges connected to it. If the network edges have weights (which might represent things like the intensity of a relationship, frequency of

> interactions, or some other quantity), the strength of the node is the sum of the weights of the edges attached to it. If the edges are unweighted, strength is simply the same as degree.

Pearson Correlation Coefficient:

> This is a statistical measure that quantifies the linear relationship between two variables. The Pearson correlation coefficient (denoted as r) ranges from -1 to +1:

- $r = 1$ indicates a perfect positive linear relationship.
- $r = -1$ indicates a perfect negative linear relationship.
- $r = 0$ indicates no linear relationship.

In [15]:
```python
from scipy.stats import pearsonr

# Function to calculate Pearson correlation for undirected and directed networks
def calculate_correlation(G):
    if G.is_directed():
        # For directed networks, calculate in-degree/out-degree and in-strength/
        in_degrees = np.array([G.in_degree(n) for n in G.nodes()])
        out_degrees = np.array([G.out_degree(n) for n in G.nodes()])

        in_strengths = np.array([sum(weight if weight is not None else 1 for _,
        out_strengths = np.array([sum(weight if weight is not None else 1 for _,
        # if weight is None, default to 1

        in_degree_corr, _ = pearsonr(in_degrees, in_strengths)
        out_degree_corr, _ = pearsonr(out_degrees, out_strengths)

        return {'In-degree vs In-strength': in_degree_corr, 'Out-degree vs Out-s

    else:
        # For undirected networks, calculate degree and strength correlations
        degrees = np.array([G.degree(n) for n in G.nodes()])
        # Correcting the sum to access the 'weight' attribute from the edge data
        strengths = np.array([sum(weight if weight is not None else 1 for _, _,
        # if weight is None, default to 1

        degree_corr, _ = pearsonr(degrees, strengths)

        return {'Degree vs Strength': degree_corr}
```

In [16]:
```python
# Iterate over each dataset
results = []
for idx, row in df.iterrows():
    fname = row['File']
    print(f"Processing {idx}...")

    if 'graphml' in fname:
        G = nx.read_graphml(fname)
    else:
        graph_class = nx.DiGraph() if 'D' in row['Type'] else nx.Graph()
        data_spec = [('weight', float)] if 'W' in row['Type'] else False
        G = nx.read_edgelist(fname, create_using=graph_class, data=data_spec)

    # Check if the graph is a multigraph
```

```
    if G.is_multigraph():
        MG = G
        G = nx.DiGraph() if MG.is_directed() else nx.Graph()
        G.add_edges_from((u,v) for u,v,i in MG.edges)

    # Calculate the correlation based on network type (directed or undirected)
    correlation_result = calculate_correlation(G)
    results.append((idx, correlation_result))
```

```
Processing Facebook Northwestern University...
Processing IMDB movies and actors...
Processing IMDB actors costar...
Processing Twitter US politics...
Processing Enron Email...
Processing Enron Executive Email...
Processing Wikipedia math...
Processing Internet routers...
Processing US air transportation...
Processing World air transportation...
Processing Yeast protein interactions...
Processing C. elegans brain...
Processing Everglades ecological food web...
```

In [17]:

```
# Display results
for result in results:
    print(f"{result[0]}: {result[1]}")
```

```
Facebook Northwestern University: {'Degree vs Strength': 1.0}
IMDB movies and actors: {'Degree vs Strength': 1.0}
IMDB actors costar: {'Degree vs Strength': 0.890951913634323}
Twitter US politics: {'In-degree vs In-strength': 0.14104902067890782, 'Out-degre
e vs Out-strength': 0.9658013120061573}
Enron Email: {'In-degree vs In-strength': 0.3850378788906514, 'Out-degree vs Out-
strength': 0.545429094848893}
Enron Executive Email: {'Degree vs Strength': 1.0}
Wikipedia math: {'In-degree vs In-strength': 0.20953020916663392, 'Out-degree vs
Out-strength': 0.999999999999945}
Internet routers: {'Degree vs Strength': 1.0}
US air transportation: {'Degree vs Strength': 0.999999999999999}
World air transportation: {'Degree vs Strength': 0.999999745813536}
Yeast protein interactions: {'Degree vs Strength': 0.9981205149459914}
C. elegans brain: {'In-degree vs In-strength': 0.3339569358740252, 'Out-degree vs
Out-strength': 0.7325317564118867}
Everglades ecological food web: {'In-degree vs In-strength': 0.09684172152672176,
'Out-degree vs Out-strength': -0.003921986547085142}
```

In [18]:

```
# Function to categorize Pearson correlation coefficient
def interpret_correlation(value):
    if value == 1.0:
        return "Perfect & positive "
    elif value == -1.0:
        return "Perfect & negative"
    elif 0.8 <= value < 1.0:
        return "Strong & positive"
    elif -1.0 < value <= -0.8:
        return "Strong & negative"
    elif 0.3 <= value < 0.8:
        return "Moderate & positive"
    elif -0.8 < value <= -0.3:
        return "Moderate & negative"
    elif -0.3 < value < 0.3:
```

```
        return "Weak or no"
    else:
        return "None"

# Print headers for better readability
print(f"{'Dataset':<40} {'Metric':<25}\t{'Value':<25} {'Linear Relationship'}")
print("="*140)

# Iterate over the results and format the output
for dataset, metrics in results:
    for metric, value in metrics.items():
        interpretation = interpret_correlation(value)
        print(f"{dataset:<40} {metric:<25}\t{value:<25} {interpretation}")
    print()
```

```
Dataset                                 Metric                         Value
Linear Relationship
=================================================================================
===========================================================
Facebook Northwestern University        Degree vs Strength             1.0
Perfect & positive

IMDB movies and actors                  Degree vs Strength             1.0
Perfect & positive

IMDB actors costar                      Degree vs Strength             0.8909519
13634323         Strong & positive

Twitter US politics                     In-degree vs In-strength       0.1410490
2067890782       Weak or no
Twitter US politics                     Out-degree vs Out-strength     0.9658013
120061573        Strong & positive

Enron Email                             In-degree vs In-strength       0.3850378
788906514        Moderate & positive
Enron Email                             Out-degree vs Out-strength     0.5454290
94848893         Moderate & positive

Enron Executive Email                   Degree vs Strength             1.0
Perfect & positive

Wikipedia math                          In-degree vs In-strength       0.2095302
0916663392       Weak or no
Wikipedia math                          Out-degree vs Out-strength     0.9999999
99999945         Strong & positive

Internet routers                        Degree vs Strength             1.0
Perfect & positive

US air transportation                   Degree vs Strength             0.9999999
99999999         Strong & positive

World air transportation                Degree vs Strength             0.9999997
45813536         Strong & positive

Yeast protein interactions              Degree vs Strength             0.9981205
149459914        Strong & positive

C. elegans brain                        In-degree vs In-strength       0.3339569
358740252        Moderate & positive
C. elegans brain                        Out-degree vs Out-strength     0.7325317
564118867        Moderate & positive

Everglades ecological food web          In-degree vs In-strength       0.0968417
2152672176       Weak or no
Everglades ecological food web          Out-degree vs Out-strength     -0.003921
986547085142     Weak or no
```