In [1]:
```
import pandas as pd
import numpy as np
import scipy.stats as stats

df=pd.read_csv('./Metadata-E2.csv')
df
```

Out[1]:

	Name	Mean	SD	Chart-Category	Chart-type	Domain	Source	Topic	Sentiment
0	5	3.06	2.045362	Area	Proportional	News	Wall Street Journal	Sports	3
1	6	-3.15	2.585400	Map	Plotted	Government	Los Angeles Planning Comission	Traffic	3
2	7	5.86	2.234405	Diagram	Isotype	Media	FiveThirtyEight	Politics	2
3	13	-5.69	2.435075	Map	Plotted	Infographics	Tableau Public	Demographics	3
4	14	-6.03	2.190402	Line	Line	Media	Envisioning Information	Economics	3
...	...	...	...	...	...	...	...	...	...
95	469	-5.17	2.050670	Diagram	Illustration	Blog	Nightingale	Geography	2
96	479	-3.19	2.062251	Network	Graph	Blog	Nightingale	Politics	3
97	481	1.03	2.071272	Area + Bar + Diagram	Stacked + Bar + Isotype	Infographics	Information is Beautiful	Lifestyle	2
98	485	-5.65	2.035646	Diagram	Illustration	Blog	Nightingale	Geology	2
99	487	5.38	2.015602	Circle	Sector	Blog	Nightingale	Health	2

100 rows × 42 columns

In [2]:
```
# Regression Results for Ratings Scores in Expt 1

import scipy.stats as stats

import csv
from matplotlib import pyplot as plt
import numpy as np
import statistics
import pandas as pd

import sklearn
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

import statsmodels.api as sm
from statsmodels.sandbox.regression.predstd import wls_prediction_std

from sklearn.decomposition import FactorAnalysis, PCA
from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import KFold
from sklearn.linear_model import BayesianRidge

from numpy import mean
from numpy import var
from math import sqrt
```

```
def cohend(d1, d2):
    n1, n2 = len(d1), len(d2)
    s1, s2 = var(d1, ddof=1), var(d2, ddof=1)
    s = sqrt(((n1 - 1) * s1 + (n2 - 1) * s2) / (n1 + n2 - 2))
    u1, u2 = mean(d1), mean(d2)
    return (u1 - u2) / s
```

In [3]:

```
# Chart-type

# Isotype, Chernoff, Illustrations
y_val=[]
for i in range(len(df)):
    if('Isotype' in df.iloc[i]['Chart-type'] or 'Chernoff' in df.iloc[i]['Chart-type'] o
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Isotype, Chernoff, Illustration : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************

# Area
y_val=[]
for i in range(len(df)):
    if('Area' in df.iloc[i]['Chart-Category']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Area : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************

# Map
y_val=[]
for i in range(len(df)):
    if('Map' in df.iloc[i]['Chart-Category']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Map : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
```

```
print("*****************************************************************************")

# Map with low annotation
y_val=[]
for i in range(len(df)):
    if('Map' in df.iloc[i]['Chart-Category'] and df.iloc[i]['Text-Vol']==1):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Map, Low Annotation : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*****************************************************************************")

# Radial Charts (chord diagrams, donut charts, timelines, radial bar charts, etc.)
y_val=[]
for i in range(len(df)):
    if('Circle' in df.iloc[i]['Chart-Category'] or 'Radial' in df.iloc[i]['Chart-type'])
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Radial Charts : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*****************************************************************************")

# Bar
y_val=[]
for i in range(len(df)):
    if('Bar' in df.iloc[i]['Chart-Category']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Bar : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*****************************************************************************")

# Point
y_val=[]
for i in range(len(df)):
    if('Point' in df.iloc[i]['Chart-Category']):
        y_val.append(1)
    else:
        y_val.append(0)
```

```
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Point : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Line
y_val=[]
for i in range(len(df)):
    if('Line' in df.iloc[i]['Chart-Category']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Line : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Distribution
y_val=[]
for i in range(len(df)):
    if('Distribution' in df.iloc[i]['Chart-Category']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Distribution : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Grid
y_val=[]
for i in range(len(df)):
    if('Grid' in df.iloc[i]['Chart-Category']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Grid : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
```

```
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Network
y_val=[]
for i in range(len(df)):
    if('Network' in df.iloc[i]['Chart-Category']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Network : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Table
y_val=[]
for i in range(len(df)):
    if('Table' in df.iloc[i]['Chart-Category']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Table : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Tree
y_val=[]
for i in range(len(df)):
    if('Tree' in df.iloc[i]['Chart-Category']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Tree : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")
```

```
Isotype, Chernoff, Illustration :
p-value : 0.00
R2 : 0.84
Cohen's d : 0.92
*********************************************************************************
*********
```

Area :
p-value : 0.00
R2 : 0.79
Cohen's d : 0.80
****
Map :
p-value : 0.00
R2 : 0.76
Cohen's d : 0.77
****
Map, Low Annotation :
p-value : 0.00
R2 : 0.80
Cohen's d : 0.81
****
Radial Charts :
p-value : 0.00
R2 : 0.72
Cohen's d : 0.65
****
Bar :
p-value : 0.00
R2 : 0.64
Cohen's d : 0.66
****
Point :
p-value : 0.00
R2 : 0.71
Cohen's d : 0.65
****
Line :
p-value : 0.00
R2 : 0.60
Cohen's d : 0.57
****
Distribution :
p-value : 0.00
R2 : 0.63
Cohen's d : 0.48
****
Grid :
p-value : 0.00
R2 : 0.68
Cohen's d : 0.69
****
Network :
p-value : 0.00
R2 : 0.72
Cohen's d : 0.63
****
Table :
p-value : 0.00
R2 : 0.43
Cohen's d : 0.37
****

```
Tree :
p-value : 0.00
R2 : 0.54
Cohen's d : 0.46
**********************************************************************************
*********
```

In [4]:

```
# Photorealistic

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Photorealistic']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Photorealistic : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("**********************************************************************************

# Human Recognizable Objects

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['HRO']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Human Recognizable Objects : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("**********************************************************************************

# Human Depiction

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['HD']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Human Depiction : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
```

```
print("*********************************************************************************")

# Skeuomorphism

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Skeumorphic']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Skeuomorphism : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Title

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Title']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Title : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Axis

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Axes']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Axis : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Axis with Tick Labels

y_val=[]
for i in range(len(df)):
```

```
    if('y' in df.iloc[i]['Axes'] and df.iloc[i]['Mean']>3):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Axis with Tick Labels : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Key
y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Key']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Key : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Title and Key

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Title'] and 'y' in df.iloc[i]['Key']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Title and Key : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Text Volume

X=df['Mean']
Y=df['Text-Vol']
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Text Volume : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
```

```
print("Cohen's d : %.2f"%d)
print("*****************************************************************************")

# Data Labels
y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Labels']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Data Labels : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*****************************************************************************")

# Data Ink Ratio

X=df['Mean']
Y=df['DIR']
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Data Ink Ratio : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*****************************************************************************")


# Visual Density

X=df['Mean']
Y=df['Density']
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Visual Density : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*****************************************************************************")

# Data Scale

X=df['Mean']
Y=df['Data-Vol']
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Data Scale : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*****************************************************************************")

# Message Redundancy
```

```
y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Msg-Red']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Message Redundancy : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Data Redundancy

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Data-Red']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Data Redundancy : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Highlights

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Highlights']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Highlights : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Arrows

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Arrows']):
        y_val.append(1)
    else:
```

```
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Arrows : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Color 7+

y_val=[]
for i in range(len(df)):
    if('low' in df.iloc[i]['Color'] or 'medium' in df.iloc[i]['Color']):
        y_val.append(0)
    else:
        y_val.append(1)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Color 7+ : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Color 2-6

y_val=[]
for i in range(len(df)):
    if('low' in df.iloc[i]['Color'] or 'high' in df.iloc[i]['Color']):
        y_val.append(0)
    else:
        y_val.append(1)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Color 2-6 : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Color 1 (black/white)

y_val=[]
for i in range(len(df)):
    if('n' in df.iloc[i]['BW']):
        y_val.append(0)
    else:
        y_val.append(1)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
```

```
results = model.fit()
print('Color 1, black and white : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")


# Background Color
X=df['Mean']
Y=df['BGC']
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Background Color : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# 3D

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['3D']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('3D : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Gridlines

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Gridlines']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Gridlines : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*********************************************************************************")

# Caption

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Caption']):
```

```
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Gridlines : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*****************************************************************************************")

# Data Source

y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Source.1']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Data Source : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*****************************************************************************************")

# Dimensionality
y_val=[]
for i in range(len(df)):
    if('low' in df.iloc[i]['Dim']):
        y_val.append(1)
    elif('medium' in df.iloc[i]['Dim']):
        y_val.append(2)
    else:
        y_val.append(3)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Dimensionality : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("*****************************************************************************************")

# Pictorial Units
y_val=[]
for i in range(len(df)):
    if('y' in df.iloc[i]['Pictorial-Units']):
        y_val.append(1)
    else:
        y_val.append(0)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
```

```
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Pictorial Units : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
print("**********************************************************************************")

# Domain
y_val=[]
for i in range(len(df)):
    if('Infographics' in df.iloc[i]['Domain']):
        y_val.append(1)
    elif('Blog' in df.iloc[i]['Domain']):
        y_val.append(2)
    elif('Government' in df.iloc[i]['Domain']):
        y_val.append(3)
    elif('News' in df.iloc[i]['Domain']):
        y_val.append(4)
    elif('Media' in df.iloc[i]['Domain']):
        y_val.append(5)
    elif('Scientific' in df.iloc[i]['Domain']):
        y_val.append(6)
    else:
        y_val.append(7)
X=df['Mean']
Y=y_val
d=cohend(Y,X)
X=sm.add_constant(X)
model = sm.OLS(Y,X)
results = model.fit()
print('Domain : ')
print("p-value : %.2f"%results.pvalues[0])
print("R2 : %.2f"%results.rsquared)
print("Cohen's d : %.2f"%d)
```

```
Photorealistic :
p-value : 0.00
R2 : 0.82
Cohen's d : 0.83
**********************************************************************************
*********
Human Recognizable Objects :
p-value : 0.00
R2 : 0.87
Cohen's d : 0.82
**********************************************************************************
*********
Human Depiction :
p-value : 0.00
R2 : 0.81
Cohen's d : 0.79
**********************************************************************************
*********
Skeuomorphism :
p-value : 0.00
R2 : 0.71
Cohen's d : 0.72
**********************************************************************************
*********
Title :
p-value : 0.00
R2 : 0.83
Cohen's d : 0.86
**********************************************************************************
```

****
Axis :
p-value : 0.00
R2 : 0.87
Cohen's d : 0.89
****
****
Axis with Tick Labels:
p-value : 0.00
R2 : 0.86
Cohen's d : 0.91
****
****
Key :
p-value : 0.00
R2 : 0.77
Cohen's d : 0.69
****
****
Title and Key :
p-value : 0.00
R2 : 0.78
Cohen's d : 0.73
****
****
Text Volume :
p-value : 0.00
R2 : 0.92
Cohen's d : 0.85
****
****
Data Labels :
p-value : 0.00
R2 : 0.74
Cohen's d : 0.83
****
****
Data Ink Ratio :
p-value : 0.00
R2 : 0.62
Cohen's d : 0.47
****
****
Visual Density :
p-value : 0.00
R2 : 0.70
Cohen's d : 0.61
****
****
Data Scale :
p-value : 0.00
R2 : 0.05
Cohen's d : 0.11
****
****
Message Redundancy :
p-value : 0.00
R2 : 0.79
Cohen's d : 0.86
****
****
Data Redundancy :
p-value : 0.00
R2 : 0.18
Cohen's d : 0.31
****

****
Highlights :
p-value : 0.00
R2 : 0.77
Cohen's d : 0.68
****
****
Arrows :
p-value : 0.00
R2 : 0.72
Cohen's d : 0.65
****
****
Color 7+ :
p-value : 0.00
R2 : 0.73
Cohen's d : 0.72
****
****
Color 2-6 :
p-value : 0.00
R2 : 0.71
Cohen's d : 0.70
****
****
Color 1, black and white :
p-value : 0.00
R2 : 0.65
Cohen's d : 0.68
****
****
Background Color :
p-value : 0.00
R2 : 0.83
Cohen's d : 0.79
****
****
3D :
p-value : 0.00
R2 : 0.56
Cohen's d : 0.33
****
****
Gridlines :
p-value : 0.00
R2 : 0.44
Cohen's d : 0.29
****
****
Caption :
p-value : 0.00
R2 : 0.68
Cohen's d : 0.62
****
****
Data Source :
p-value : 0.00
R2 : 0.07
Cohen's d : 0.18
****
****
Dimensionality :
p-value : 0.00
R2 : 0.06
Cohen's d : 0.14
****

```
*********
Pictorial Units :
p-value : 0.00
R2 : 0.77
Cohen's d : 0.82
********************************************************************************
*********
Domain :
p-value : 0.00
R2 : 0.12
Cohen's d : 0.15
```

In [ ]:

In [ ]: