```
% import data
data = readtable("impact.xlsx")
```

```
Warning: Column headers from the file were modified to make them valid MATLAB identifiers
before creating variable names for the table. The original column headers are saved in the
VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column headers as table variable names.
data = 300×52 table
```

...

	Horodateur	Gender	Age	Education_Level	Department
1	17-Nov-2024 17:19:20	'Male'	'18-25'	'PhD'	'Sales and Marketing'
2	17-Nov-2024 17:19:20	'Male'	'56 and above'	'Diploma'	'Sales and Marketing'
3	17-Nov-2024 17:19:20	'Other'	'46-55'	'Masters'	'Sales and Marketing'
4	17-Nov-2024 17:19:20	'Other'	'56 and above'	'PhD'	'Sales and Marketing'
5	17-Nov-2024 17:19:20	'Other'	'26-35'	'Degree'	'Engineering'
6	17-Nov-2024 17:19:20	'Female'	'26-35'	'Degree'	'Sales and Marketing'
7	17-Nov-2024 17:19:20	'Other'	'18-25'	'PhD'	'HR and Admin'
8	17-Nov-2024 17:19:20	'Female'	'56 and above'	'Degree'	'Engineering'
9	17-Nov-2024 17:19:20	'Other'	'36-45'	'PhD'	'Engineering'
10	17-Nov-2024 17:19:20	'Other'	'46-55'	'PhD'	'Logistics and Supply'
11	17-Nov-2024 17:19:20	'Male'	'36-45'	'Diploma'	'Logistics and Supply'
12	17-Nov-2024 17:19:20	'Male'	'36-45'	'PhD'	'Sales and Marketing'
13	17-Nov-2024 17:19:20	'Female'	'18-25'	'Degree'	'Sales and Marketing'
14	17-Nov-2024 17:19:20	'Other'	'26-35'	'Degree'	'Operations'
15	17-Nov-2024 17:19:20	'Other'	'56 and above'	'Diploma'	'Sales and Marketing'
16	17-Nov-2024 17:19:20	'Other'	'46-55'	'Masters'	'HR and Admin'
17	17-Nov-2024 17:19:20	'Other'	'26-35'	'Diploma'	'HR and Admin'
18	17-Nov-2024 17:19:20	'Male'	'36-45'	'Diploma'	'Sales and Marketing'
19	17-Nov-2024 17:19:20	'Other'	'36-45'	'PhD'	'Operations'
20	17-Nov-2024 17:19:20	'Other'	'18-25'	'PhD'	'Operations'
21	17-Nov-2024 17:19:20	'Male'	'46-55'	'Degree'	'Operations'
22	17-Nov-2024 17:19:20	'Female'	'26-35'	'Degree'	'Sales and Marketing'
23	17-Nov-2024 17:19:20	'Female'	'26-35'	'PhD'	'Operations'
24	17-Nov-2024 17:19:20	'Female'	'36-45'	'Masters'	'Sales and Marketing'
25	17-Nov-2024 17:19:20	'Male'	'56 and above'	'Degree'	'Operations'
26	17-Nov-2024 17:19:20	'Female'	'18-25'	'Masters'	'Logistics and Supply'
27	17-Nov-2024 17:19:20	'Male'	'26-35'	'Diploma'	'Operations'
28	17-Nov-2024 17:19:20	'Female'	'56 and above'	'PhD'	'Engineering'

	Horodateur	Gender	Age	Education_Level	Department
29	17-Nov-2024 17:19:20	'Other'	'36-45'	'Degree'	'Logistics and Supply'
30	17-Nov-2024 17:19:20	'Other'	'56 and above'	'Masters'	'HR and Admin'
31	17-Nov-2024 17:19:20	'Male'	'36-45'	'Diploma'	'Operations'
32	17-Nov-2024 17:19:20	'Female'	'46-55'	'Masters'	'Sales and Marketing'
33	17-Nov-2024 17:19:20	'Male'	'56 and above'	'Degree'	'Engineering'
34	17-Nov-2024 17:19:20	'Other'	'18-25'	'Masters'	'Sales and Marketing'
35	17-Nov-2024 17:19:20	'Male'	'18-25'	'Diploma'	'Engineering'
36	17-Nov-2024 17:19:20	'Male'	'56 and above'	'Diploma'	'HR and Admin'
37	17-Nov-2024 17:19:20	'Male'	'46-55'	'Masters'	'Logistics and Supply'
38	17-Nov-2024 17:19:20	'Female'	'18-25'	'PhD'	'Operations'
39	17-Nov-2024 17:19:20	'Other'	'36-45'	'PhD'	'Sales and Marketing'
40	17-Nov-2024 17:19:20	'Female'	'46-55'	'Diploma'	'Engineering'
41	17-Nov-2024 17:19:20	'Other'	'18-25'	'Masters'	'Engineering'
42	17-Nov-2024 17:19:20	'Female'	'18-25'	'Degree'	'Engineering'
43	17-Nov-2024 17:19:20	'Female'	'18-25'	'Diploma'	'Logistics and Supply'
44	17-Nov-2024 17:19:20	'Other'	'46-55'	'PhD'	'Operations'
45	17-Nov-2024 17:19:20	'Male'	'18-25'	'Diploma'	'HR and Admin'
46	17-Nov-2024 17:19:20	'Other'	'56 and above'	'Degree'	'Sales and Marketing'
47	17-Nov-2024 17:19:20	'Other'	'36-45'	'Degree'	'Engineering'
48	17-Nov-2024 17:19:20	'Male'	'56 and above'	'Masters'	'Sales and Marketing'
49	17-Nov-2024 17:19:20	'Female'	'56 and above'	'PhD'	'Engineering'
50	17-Nov-2024 17:19:20	'Other'	'26-35'	'PhD'	'Operations'
51	17-Nov-2024 17:19:20	'Other'	'46-55'	'PhD'	'Operations'
52	17-Nov-2024 17:19:20	'Other'	'18-25'	'Masters'	'Engineering'
53	17-Nov-2024 17:19:20	'Female'	'56 and above'	'PhD'	'Logistics and Supply'
54	17-Nov-2024 17:19:20	'Other'	'36-45'	'Degree'	'HR and Admin'
55	17-Nov-2024 17:19:20	'Female'	'18-25'	'Diploma'	'Sales and Marketing'
56	17-Nov-2024 17:19:20	'Other'	'46-55'	'PhD'	'HR and Admin'
57	17-Nov-2024 17:19:20	'Female'	'26-35'	'Degree'	'HR and Admin'
58	17-Nov-2024 17:19:20	'Other'	'26-35'	'Masters'	'Engineering'
59	17-Nov-2024 17:19:20	'Female'	'46-55'	'PhD'	'Engineering'
60	17-Nov-2024 17:19:20	'Female'	'36-45'	'PhD'	'HR and Admin'
61	17-Nov-2024 17:19:20	'Female'	'18-25'	'Degree'	'Operations'

	Horodateur	Gender	Age	Education_Level	Department
62	17-Nov-2024 17:19:20	'Female'	'18-25'	'PhD'	'Sales and Marketing'
63	17-Nov-2024 17:19:20	'Male'	'26-35'	'Degree'	'Operations'
64	17-Nov-2024 17:19:20	'Other'	'36-45'	'PhD'	'Engineering'
65	17-Nov-2024 17:19:20	'Other'	'18-25'	'Masters'	'Sales and Marketing'
66	17-Nov-2024 17:19:20	'Other'	'56 and above'	'Degree'	'Engineering'
67	17-Nov-2024 17:19:20	'Male'	'26-35'	'PhD'	'Operations'
68	17-Nov-2024 17:19:20	'Other'	'46-55'	'PhD'	'Sales and Marketing'
69	17-Nov-2024 17:19:20	'Female'	'46-55'	'Masters'	'Operations'
70	17-Nov-2024 17:19:20	'Male'	'36-45'	'Degree'	'HR and Admin'
71	17-Nov-2024 17:19:20	'Other'	'26-35'	'Diploma'	'Operations'
72	17-Nov-2024 17:19:20	'Other'	'46-55'	'PhD'	'HR and Admin'
73	17-Nov-2024 17:19:20	'Female'	'18-25'	'Diploma'	'HR and Admin'
74	17-Nov-2024 17:19:20	'Other'	'26-35'	'Masters'	'Engineering'
75	17-Nov-2024 17:19:20	'Female'	'36-45'	'PhD'	'Engineering'
76	17-Nov-2024 17:19:20	'Other'	'46-55'	'Masters'	'Engineering'
77	17-Nov-2024 17:19:20	'Male'	'56 and above'	'Degree'	'Operations'
78	17-Nov-2024 17:19:20	'Other'	'36-45'	'Degree'	'Engineering'
79	17-Nov-2024 17:19:20	'Female'	'26-35'	'Diploma'	'Sales and Marketing'
80	17-Nov-2024 17:19:20	'Male'	'36-45'	'Masters'	'HR and Admin'
81	17-Nov-2024 17:19:20	'Male'	'18-25'	'Degree'	'Sales and Marketing'
82	17-Nov-2024 17:19:20	'Female'	'56 and above'	'Diploma'	'Sales and Marketing'
83	17-Nov-2024 17:19:20	'Female'	'46-55'	'Diploma'	'Engineering'
84	17-Nov-2024 17:19:20	'Male'	'36-45'	'Degree'	'Logistics and Supply'
85	17-Nov-2024 17:19:20	'Other'	'18-25'	'PhD'	'HR and Admin'
86	17-Nov-2024 17:19:20	'Male'	'18-25'	'Masters'	'Engineering'
87	17-Nov-2024 17:19:20	'Female'	'26-35'	'Diploma'	'Operations'
88	17-Nov-2024 17:19:20	'Other'	'18-25'	'PhD'	'Sales and Marketing'
89	17-Nov-2024 17:19:20	'Other'	'46-55'	'Diploma'	'Logistics and Supply'
90	17-Nov-2024 17:19:20	'Male'	'46-55'	'PhD'	'Logistics and Supply'
91	17-Nov-2024 17:19:20	'Other'	'46-55'	'Diploma'	'Sales and Marketing'
92	17-Nov-2024 17:19:20	'Other'	'26-35'	'Masters'	'Engineering'
93	17-Nov-2024 17:19:20	'Other'	'18-25'	'PhD'	'Engineering'
94	17-Nov-2024 17:19:20	'Other'	'46-55'	'PhD'	'Operations'

	Horodateur	Gender	Age	Education_Level	Department
95	17-Nov-2024 17:19:20	'Female'	'46-55'	'Diploma'	'Sales and Marketing'
96	17-Nov-2024 17:19:20	'Female'	'18-25'	'Degree'	'Operations'
97	17-Nov-2024 17:19:20	'Male'	'26-35'	'Degree'	'Logistics and Supply'
98	17-Nov-2024 17:19:20	'Male'	'46-55'	'PhD'	'Sales and Marketing'
99	17-Nov-2024 17:19:20	'Female'	'46-55'	'Diploma'	'Engineering'
100	17-Nov-2024 17:19:20	'Female'	'56 and above'	'Masters'	'Logistics and Supply'

⋮

```
% Extract the Gender column
genderData = data.Gender
```

```
genderData = 300x1 cell
'Male'
'Male'
'Other'
'Other'
'Other'
'Female'
'Other'
'Female'
'Other'
'Other'
    ⋮
```

```
% Count the occurrences of each gender
genderCounts = groupcounts(categorical(genderData))
```

```
genderCounts = 3×1
    97
    94
   109
```

```
% Calculate the total number of entries
totalEntries = sum(genderCounts)
```

```
totalEntries =
300
```

```
% Calculate the percentage for each gender
genderPercentages = (genderCounts / totalEntries) * 100
```

```
genderPercentages = 3×1
   32.3333
   31.3333
   36.3333
```

```
% Display the results
disp('Gender Counts:');
```

```
Gender Counts:
```

```
disp(genderCounts);
```

```
    97
    94
   109
```

```
disp('Gender Percentages:');
```

```
Gender Percentages:
```

```
disp(genderPercentages);
```

```
   32.3333
   31.3333
   36.3333
```

```
% Optional: If gender labels are not automatically sorted, map them
genderLabels = categories(categorical(genderData));

% Display results with labels
for i = 1:length(genderLabels)
    fprintf('%s: %.2f%%\n', genderLabels{i}, genderPercentages(i));
end
```

```
Female: 32.33%
Male: 31.33%
Other: 36.33%
```

```
% Extract the Gender column
ageData = data.Age;

% Count the occurrences of each age
ageCounts = groupcounts(categorical(ageData));

% Calculate the total number of entries
totalEntries = sum(ageCounts);

% Calculate the percentage for each age
agePercentages = (ageCounts / totalEntries) * 100;

% Display the results
disp('Age Counts:');
```

```
Age Counts:
```

```
disp(ageCounts);
```

```
    71
    71
    54
    51
    53
```

```
disp('Age Percentages:');
```

Age Percentages:

```
disp(agePercentages);
```

```
   23.6667
   23.6667
   18.0000
   17.0000
   17.6667
```

```
% Optional: If age labels are not automatically sorted, map them
ageLabels = categories(categorical(ageData));

% Display results with labels
for i = 1:length(ageLabels)
    fprintf('%s: %.2f%%\n', ageLabels{i}, agePercentages(i));
end
```

```
18-25: 23.67%
26-35: 23.67%
36-45: 18.00%
46-55: 17.00%
56 and above: 17.67%
```

```
% Extract the Education_Level column
education_LevelData = data.Education_Level;

% Count the occurrences of each education_Level
education_LevelCounts = groupcounts(categorical(education_LevelData));

% Calculate the total number of entries
totalEntries = sum(education_LevelCounts);

% Calculate the percentage for each education_Level
education_LevelPercentages = (education_LevelCounts / totalEntries) * 100;

% Display the results
disp('Education_Level Counts:');
```

Education_Level Counts:

```
disp(education_LevelCounts);
```

```
    78
    65
    71
    86
```

```
disp('Education_Level Percentages:');
```

Education_Level Percentages:

```
disp(education_LevelPercentages);
```

```
   26.0000
   21.6667
   23.6667
   28.6667
```

```
% Optional: If education_Level labels are not automatically sorted, map them
education_LevelLabels = categories(categorical(education_LevelData));

% Display results with labels
for i = 1:length(education_LevelLabels)
    fprintf('%s: %.2f%%\n', education_LevelLabels{i},
education_LevelPercentages(i));
end
```

```
Degree: 26.00%
Diploma: 21.67%
Masters: 23.67%
PhD: 28.67%
```

```
% Extract the Department column
departmentData = data.Department;

% Count the occurrences of each department
departmentCounts = groupcounts(categorical(departmentData));

% Calculate the total number of entries
totalEntries = sum(departmentCounts);

% Calculate the percentage for each department
departmentPercentages = (departmentCounts / totalEntries) * 100;

% Display the results
disp('Department Counts:');
```

```
Department Counts:
```

```
disp(departmentCounts);
```

```
    60
    57
    56
    68
    59
```

```
disp('Department Percentages:');
```

```
Department Percentages:
```

```
disp(departmentPercentages);
```

```
   20.0000
   19.0000
   18.6667
```

```
   22.6667
   19.6667
```

```
% Optional: If gender labels are not automatically sorted, map them
departmentLabels = categories(categorical(departmentData));

% Display results with labels
for i = 1:length(departmentLabels)
    fprintf('%s: %.2f%%\n', departmentLabels{i}, departmentPercentages(i));
end
```

```
Engineering: 20.00%
HR and Admin: 19.00%
Logistics and Supply: 18.67%
Operations: 22.67%
Sales and Marketing: 19.67%
```

```
% Extract the Position column
positionData = data.Position;

% Count the occurrences of each position
positionCounts = groupcounts(categorical(positionData));

% Calculate the total number of entries
totalEntries = sum(positionCounts);

% Calculate the percentage for each position
positionPercentages = (positionCounts / totalEntries) * 100;

% Display the results
disp('Position Counts:');
```

```
Position Counts:
```

```
disp(positionCounts);
```

```
    77
    69
    77
    77
```

```
disp('Position Percentages:');
```

```
Position Percentages:
```

```
disp(positionPercentages);
```

```
   25.6667
   23.0000
   25.6667
   25.6667
```

```
% Optional: If position labels are not automatically sorted, map them
```

```
positionLabels = categories(categorical(positionData));

% Display results with labels
for i = 1:length(positionLabels)
    fprintf('%s: %.2f%%\n', positionLabels{i}, positionPercentages(i));
end
```

```
Executive/Director: 25.67%
Manager: 23.00%
Staff member/employee: 25.67%
Supervisor: 25.67%
```

```
% Extract the Service_Years column
service_YearsData = data.Service_Years;

% Count the occurrences of each service_Years
service_YearsCounts = groupcounts(categorical(service_YearsData));

% Calculate the total number of entries
totalEntries = sum(service_YearsCounts);

% Calculate the percentage for each service_Years
service_YearsPercentages = (service_YearsCounts / totalEntries) * 100;

% Display the results
disp('Service_Years Counts:');
```

```
Service_Years Counts:
```

```
disp(service_YearsCounts);
```

```
    80
    77
    69
    74
```

```
disp('Service_Years Percentages:');
```

```
Service_Years Percentages:
```

```
disp(service_YearsPercentages);
```

```
   26.6667
   25.6667
   23.0000
   24.6667
```

```
% Optional: If gender labels are not automatically sorted, map them
service_YearsLabels = categories(categorical(service_YearsData));

% Display results with labels
for i = 1:length(service_YearsLabels)
```

```
    fprintf('%s: %.2f%%\n', service_YearsLabels{i},
service_YearsPercentages(i));
end
```

```
1-3 years: 26.67%
3-5 years: 25.67%
Less than 1 year: 23.00%
More than 5 years: 24.67%
```

```
% Extract the Marital_Status column
marital_StatusData = data.Marital_Status;

% Count the occurrences of each marital_Status
marital_StatusCounts = groupcounts(categorical(marital_StatusData));

% Calculate the total number of entries
totalEntries = sum(marital_StatusCounts);

% Calculate the percentage for each marital_Status
marital_StatusPercentages = (marital_StatusCounts / totalEntries) * 100;

% Display the results
disp('Marital_Status Counts:');
```

```
Marital_Status Counts:
```

```
disp(marital_StatusCounts);
```

```
   140
   160
```

```

disp('Marital_Status Percentages:');
```

```
Marital_Status Percentages:
```

```
disp(marital_StatusPercentages);
```

```
   46.6667
   53.3333
```

```

% Optional: If marital_Status labels are not automatically sorted, map them
marital_StatusLabels = categories(categorical(marital_StatusData));

% Display results with labels
for i = 1:length(marital_StatusLabels)
    fprintf('%s: %.2f%%\n', marital_StatusLabels{i},
marital_StatusPercentages(i));
end
```

```
Married: 46.67%
Single: 53.33%
```

```
% Extract the AutocraticLeadership variables
```

```
autocraticLeadership = [data.AutocraticLeadership1,
data.AutocraticLeadership2, data.AutocraticLeadership3,
data.AutocraticLeadership4, data.AutocraticLeadership5,
data.AutocraticLeadership6, data.AutocraticLeadership7,
data.AutocraticLeadership8];

% Calculate the average for each row
averageAutocraticLeadership = mean(autocraticLeadership, 2)
```

```
averageAutocraticLeadership = 300×1
    3.6250
    2.7500
    3.2500
    3.3750
    2.5000
    2.5000
    3.1250
    2.6250
    3.0000
    4.0000
     ⋮
```

```
% Calculate summary statistics for the average autocratic leadership scores
overallMean = mean(averageAutocraticLeadership, 'omitnan'); % Mean
overallMedian = median(averageAutocraticLeadership, 'omitnan'); % Median
overallStd = std(averageAutocraticLeadership, 'omitnan'); % Standard
deviation
overallMin = min(averageAutocraticLeadership); % Minimum
overallMax = max(averageAutocraticLeadership); % Maximum

% Display the results
disp('Summary Statistics for Average Autocratic Leadership:');
```

```
Summary Statistics for Average Autocratic Leadership:
```

```
fprintf('Mean: %.2f\n', overallMean);
```

```
Mean: 3.00
```

```
fprintf('Median: %.2f\n', overallMedian);
```

```
Median: 3.00
```

```
fprintf('Standard Deviation: %.2f\n', overallStd);
```

```
Standard Deviation: 0.49
```

```
fprintf('Minimum: %.2f\n', overallMin);
```

```
Minimum: 1.75
```

```
fprintf('Maximum: %.2f\n', overallMax);
```

```
Maximum: 4.38
```

```
% Extract the Power_Distance variables
```

```
power_Distance = [data.Power_Distance1, data.Power_Distance2,
data.Power_Distance3, data.Power_Distance4];

% Calculate the average for each row
averagePower_Distance = mean(power_Distance, 2)
```

```
averagePower_Distance = 300×1
    2.5000
    3.5000
    2.7500
    2.7500
    3.0000
    3.0000
    3.0000
    2.7500
    2.2500
    2.0000
     ⋮
```

```
% Calculate summary statistics for the average power distance scores
overallMean = mean(averagePower_Distance, 'omitnan'); % Mean
overallMedian = median(averagePower_Distance, 'omitnan'); % Median
overallStd = std(averagePower_Distance, 'omitnan'); % Standard deviation
overallMin = min(averagePower_Distance); % Minimum
overallMax = max(averagePower_Distance); % Maximum

% Display the results
disp('Summary Statistics for Average Power Distance:');
```

```
Summary Statistics for Average Power Distance:
```

```
fprintf('Mean: %.2f\n', overallMean);
```

```
Mean: 3.07
```

```
fprintf('Median: %.2f\n', overallMedian);
```

```
Median: 3.00
```

```
fprintf('Standard Deviation: %.2f\n', overallStd);
```

```
Standard Deviation: 0.68
```

```
fprintf('Minimum: %.2f\n', overallMin);
```

```
Minimum: 1.00
```

```
fprintf('Maximum: %.2f\n', overallMax);
```

```
Maximum: 5.00
```

```
% Extract the IndividualismvsCollectivism variables
individualismvsCollectivism = [data.IndividualismvsCollectivism1,
data.IndividualismvsCollectivism2, data.IndividualismvsCollectivism3,
data.IndividualismvsCollectivism4];
```

```
% Calculate the average for each row
averageIndividualismvsCollectivism = mean(individualismvsCollectivism , 2)
```

```
averageIndividualismvsCollectivism = 300×1
    1.7500
    2.2500
    2.5000
    3.0000
    3.7500
    2.5000
    2.5000
    4.2500
    3.2500
    3.7500
     ⋮
```

```
% Calculate summary statistics for the average individualism vs Collectivism
scores
overallMean = mean(averageIndividualismvsCollectivism , 'omitnan'); % Mean
overallMedian = median(averageIndividualismvsCollectivism , 'omitnan'); %
Median
overallStd = std(averageIndividualismvsCollectivism , 'omitnan'); % Standard
deviation
overallMin = min(averageIndividualismvsCollectivism); % Minimum
overallMax = max(averageIndividualismvsCollectivism); % Maximum

% Display the results
disp('Summary Statistics for Average Individualism vs Collectivism :');
```

```
Summary Statistics for Average Individualism vs Collectivism :
```

```
fprintf('Mean: %.2f\n', overallMean);
```

```
Mean: 2.98
```

```
fprintf('Median: %.2f\n', overallMedian);
```

```
Median: 3.00
```

```
fprintf('Standard Deviation: %.2f\n', overallStd);
```

```
Standard Deviation: 0.70
```

```
fprintf('Minimum: %.2f\n', overallMin);
```

```
Minimum: 1.25
```

```
fprintf('Maximum: %.2f\n', overallMax);
```

```
Maximum: 4.50
```

```
% Extract the MasculinityvsFemininity variables
masculinityvsFemininity = [data.MasculinityvsFemininity1,
data.MasculinityvsFemininity2, data.MasculinityvsFemininity3,
data.MasculinityvsFemininity4];
```

```
% Calculate the average for each row
averageMasculinityvsFemininity = mean(masculinityvsFemininity, 2)
```

```
averageMasculinityvsFemininity = 300×1
    1.7500
    3.5000
    3.7500
    1.2500
    2.5000
    2.5000
    3.0000
    3.7500
    2.7500
    2.5000
     ⋮
```

```
% Calculate summary statistics for the average masculinity vs femininity
scores
overallMean = mean(averageMasculinityvsFemininity, 'omitnan'); % Mean
overallMedian = median(averageMasculinityvsFemininity, 'omitnan'); % Median
overallStd = std(averageMasculinityvsFemininity, 'omitnan'); % Standard
deviation
overallMin = min(averageMasculinityvsFemininity); % Minimum
overallMax = max(averageMasculinityvsFemininity); % Maximum

% Display the results
disp('Summary Statistics for Average Masculinity vs Femininity:');
```

```
Summary Statistics for Average Masculinity vs Femininity:
```

```
fprintf('Mean: %.2f\n', overallMean);
```

```
Mean: 2.98
```

```
fprintf('Median: %.2f\n', overallMedian);
```

```
Median: 3.00
```

```
fprintf('Standard Deviation: %.2f\n', overallStd);
```

```
Standard Deviation: 0.75
```

```
fprintf('Minimum: %.2f\n', overallMin);
```

```
Minimum: 1.25
```

```
fprintf('Maximum: %.2f\n', overallMax);
```

```
Maximum: 5.00
```

```
% Extract the UncertaintyAvoidance variables
uncertaintyAvoidance = [data.UncertaintyAvoidance1,
data.UncertaintyAvoidance2, data.UncertaintyAvoidance3,
data.UncertaintyAvoidance4];

% Calculate the average for each row
```

```
averageUncertaintyAvoidance = mean(uncertaintyAvoidance, 2)
```

```
averageUncertaintyAvoidance = 300×1
    2.5000
    3.0000
    2.5000
    1.7500
    3.5000
    2.7500
    2.5000
    1.5000
    3.0000
    3.0000
     ⋮
```

```
% Calculate summary statistics for the average uncertainty avoidance scores
overallMean = mean(averageUncertaintyAvoidance, 'omitnan'); % Mean
overallMedian = median(averageUncertaintyAvoidance, 'omitnan'); % Median
overallStd = std(averageUncertaintyAvoidance, 'omitnan'); % Standard
deviation
overallMin = min(averageUncertaintyAvoidance); % Minimum
overallMax = max(averageUncertaintyAvoidance); % Maximum

% Display the results
disp('Summary Statistics for Average Uncertainty Avoidance:');
```

```
Summary Statistics for Average Uncertainty Avoidance:
```

```
fprintf('Mean: %.2f\n', overallMean);
```

```
Mean: 3.05
```

```
fprintf('Median: %.2f\n', overallMedian);
```

```
Median: 3.00
```

```
fprintf('Standard Deviation: %.2f\n', overallStd);
```

```
Standard Deviation: 0.73
```

```
fprintf('Minimum: %.2f\n', overallMin);
```

```
Minimum: 1.25
```

```
fprintf('Maximum: %.2f\n', overallMax);
```

```
Maximum: 4.75
```

```
% Extract the Long_termvsShort_termOrientation variables
long_termvsShort_termOrientation = [data.Long_termvsShort_termOrientation1,
data.Long_termvsShort_termOrientation2,
data.Long_termvsShort_termOrientation3,
data.Long_termvsShort_termOrientation4];

% Calculate the average for each row
```

```
averageLong_termvsShort_termOrientation =
mean(long_termvsShort_termOrientation, 2)
```

```
averageLong_termvsShort_termOrientation = 300×1
    3.5000
    3.5000
    2.5000
    3.0000
    3.7500
    3.5000
    2.7500
    2.0000
    2.7500
    3.0000
     ⋮
```

```
% Calculate summary statistics for the average long_term vs
short_termOrientation scores
overallMean = mean(averageLong_termvsShort_termOrientation, 'omitnan'); %
Mean
overallMedian = median(averageLong_termvsShort_termOrientation, 'omitnan');
% Median
overallStd = std(averageLong_termvsShort_termOrientation, 'omitnan'); %
Standard deviation
overallMin = min(averageLong_termvsShort_termOrientation); % Minimum
overallMax = max(averageLong_termvsShort_termOrientation); % Maximum

% Display the results
disp('Summary Statistics for Average Long_term vs Short_termOrientation:');
```

```
Summary Statistics for Average Long_term vs Short_termOrientation:
```

```
fprintf('Mean: %.2f\n', overallMean);
```

```
Mean: 3.01
```

```
fprintf('Median: %.2f\n', overallMedian);
```

```
Median: 3.00
```

```
fprintf('Standard Deviation: %.2f\n', overallStd);
```

```
Standard Deviation: 0.73
```

```
fprintf('Minimum: %.2f\n', overallMin);
```

```
Minimum: 1.25
```

```
fprintf('Maximum: %.2f\n', overallMax);
```

```
Maximum: 4.75
```

```
% Extract the TaskPerformance variables
taskPerformance = [data.TaskPerformance1, data.TaskPerformance2,
data.TaskPerformance3, data.TaskPerformance4];
```

```
% Calculate the average for each row
averageTaskPerformance = mean(taskPerformance, 2)
```

```
averageTaskPerformance = 300×1
    2.5000
    4.0000
    2.0000
    2.5000
    3.7500
    2.7500
    3.0000
    3.7500
    3.5000
    2.2500
     ⋮
```

```
% Calculate summary statistics for the average task performance scores
overallMean = mean(averageTaskPerformance, 'omitnan'); % Mean
overallMedian = median(averageTaskPerformance, 'omitnan'); % Median
overallStd = std(averageTaskPerformance, 'omitnan'); % Standard deviation
overallMin = min(averageTaskPerformance); % Minimum
overallMax = max(averageTaskPerformance); % Maximum

% Display the results
disp('Summary Statistics for Average Task Performance:');
```

```
Summary Statistics for Average Task Performance:
```

```
fprintf('Mean: %.2f\n', overallMean);
```

```
Mean: 2.96
```

```
fprintf('Median: %.2f\n', overallMedian);
```

```
Median: 3.00
```

```
fprintf('Standard Deviation: %.2f\n', overallStd);
```

```
Standard Deviation: 0.71
```

```
fprintf('Minimum: %.2f\n', overallMin);
```

```
Minimum: 1.00
```

```
fprintf('Maximum: %.2f\n', overallMax);
```

```
Maximum: 4.75
```

```
% Extract the ContextualPerformance variables
contextualPerformance = [data.ContextualPerformance1,
data.ContextualPerformance2, data.ContextualPerformance3,
data.ContextualPerformance4];

% Calculate the average for each row
averageContextualPerformance = mean(contextualPerformance, 2)
```

```
averageContextualPerformance = 300×1
    2.0000
    2.7500
    4.5000
    3.2500
    1.7500
    3.5000
    2.7500
    3.2500
    2.5000
    3.2500
     ⋮
```

```
% Calculate summary statistics for the average contextual performance scores
overallMean = mean(averageContextualPerformance, 'omitnan'); % Mean
overallMedian = median(averageContextualPerformance, 'omitnan'); % Median
overallStd = std(averageContextualPerformance, 'omitnan'); % Standard
deviation
overallMin = min(averageContextualPerformance); % Minimum
overallMax = max(averageContextualPerformance); % Maximum

% Display the results
disp('Summary Statistics for Average Contextual Performance:');
```

```
Summary Statistics for Average Contextual Performance:
```

```
fprintf('Mean: %.2f\n', overallMean);
```

```
Mean: 3.09
```

```
fprintf('Median: %.2f\n', overallMedian);
```

```
Median: 3.25
```

```
fprintf('Standard Deviation: %.2f\n', overallStd);
```

```
Standard Deviation: 0.72
```

```
fprintf('Minimum: %.2f\n', overallMin);
```

```
Minimum: 1.25
```

```
fprintf('Maximum: %.2f\n', overallMax);
```

```
Maximum: 4.75
```

```
% Extract the AdaptivePerformance variables
adaptivePerformance = [data.AdaptivePerformance1, data.AdaptivePerformance2,
data.AdaptivePerformance3, data.AdaptivePerformance4];

% Calculate the average for each row
averageAdaptivePerformance = mean(adaptivePerformance, 2)
```

```
averageAdaptivePerformance = 300×1
    3.5000
    2.7500
```

```
    4.0000
    3.0000
    4.2500
    2.5000
    3.0000
    2.5000
    2.7500
    3.0000
     ⋮
```

```
% Calculate summary statistics for the average adaptive performance scores
overallMean = mean(averageAdaptivePerformance, 'omitnan'); % Mean
overallMedian = median(averageAdaptivePerformance, 'omitnan'); % Median
overallStd = std(averageAdaptivePerformance, 'omitnan'); % Standard deviation
overallMin = min(averageAdaptivePerformance); % Minimum
overallMax = max(averageAdaptivePerformance); % Maximum

% Display the results
disp('Summary Statistics for Average Adaptive Performance:');
```

```
Summary Statistics for Average Adaptive Performance:
```

```
fprintf('Mean: %.2f\n', overallMean);
```

```
Mean: 3.03
```

```
fprintf('Median: %.2f\n', overallMedian);
```

```
Median: 3.00
```

```
fprintf('Standard Deviation: %.2f\n', overallStd);
```

```
Standard Deviation: 0.74
```

```
fprintf('Minimum: %.2f\n', overallMin);
```

```
Minimum: 1.00
```

```
fprintf('Maximum: %.2f\n', overallMax);
```

```
Maximum: 5.00
```

```
% Extract the CounterproductiveWorkBehavior variables
counterproductiveWorkBehavior = [data.CounterproductiveWorkBehavior1,
data.CounterproductiveWorkBehavior2, data.CounterproductiveWorkBehavior3,
data.CounterproductiveWorkBehavior4];

% Calculate the average for each row
averageCounterproductiveWorkBehavior = mean(counterproductiveWorkBehavior, 2)
```

```
averageCounterproductiveWorkBehavior = 300×1
    2.2500
    3.5000
    2.0000
    2.5000
    3.0000
```

```
    2.7500
    2.7500
    2.7500
    3.0000
    2.2500
     ⋮
     ⋮
```

```
% Calculate summary statistics for the average counterproductive work
behavior scores
overallMean = mean(averageCounterproductiveWorkBehavior, 'omitnan'); % Mean
overallMedian = median(averageCounterproductiveWorkBehavior, 'omitnan'); %
Median
overallStd = std(averageCounterproductiveWorkBehavior, 'omitnan'); %
Standard deviation
overallMin = min(averageCounterproductiveWorkBehavior); % Minimum
overallMax = max(averageCounterproductiveWorkBehavior); % Maximum

% Display the results
disp('Summary Statistics for Average Counterproductive Work Behavior:');
```

```
Summary Statistics for Average Counterproductive Work Behavior:
```

```
fprintf('Mean: %.2f\n', overallMean);
```

```
Mean: 3.00
```

```
fprintf('Median: %.2f\n', overallMedian);
```

```
Median: 3.00
```

```
fprintf('Standard Deviation: %.2f\n', overallStd);
```

```
Standard Deviation: 0.71
```

```
fprintf('Minimum: %.2f\n', overallMin);
```

```
Minimum: 1.25
```

```
fprintf('Maximum: %.2f\n', overallMax);
```

```
Maximum: 5.00
```

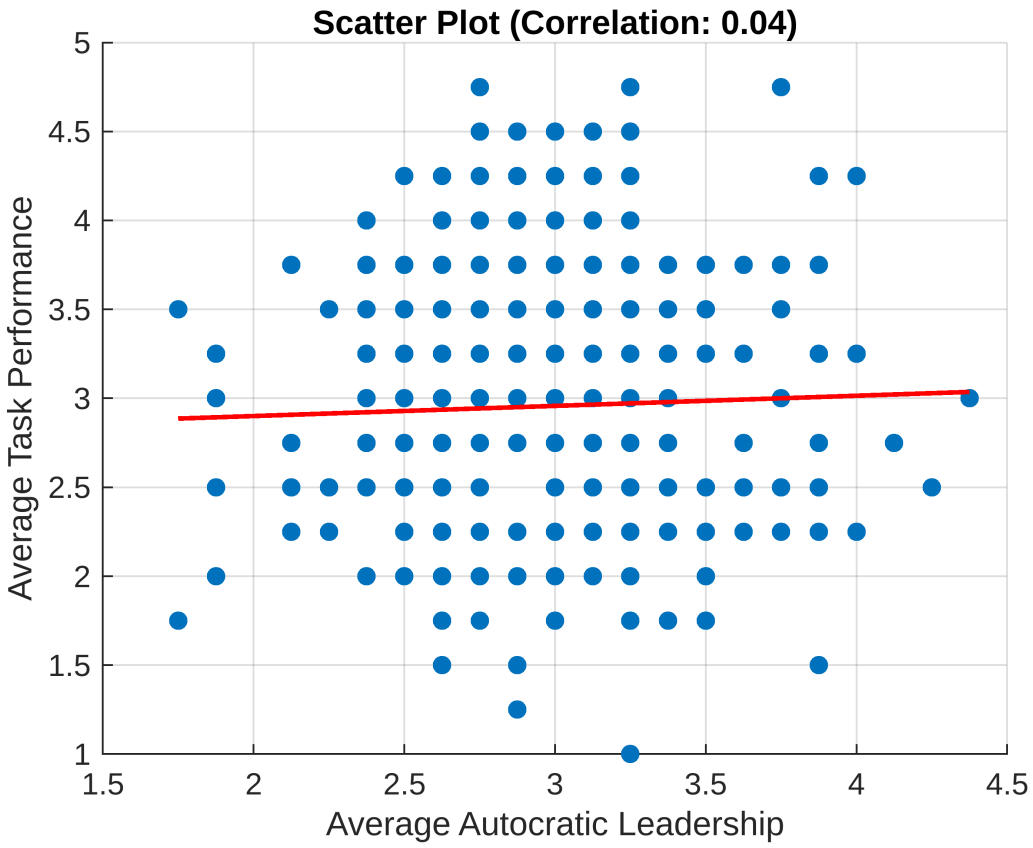
```
% Compute the correlation
correlationValue = corr(averageAutocraticLeadership, averageTaskPerformance)
```

```
correlationValue =
0.0396
```

```
% Scatter plot with trend line
scatter(averageAutocraticLeadership, averageTaskPerformance, 'filled');
hold on;
p = polyfit(averageAutocraticLeadership, averageTaskPerformance, 1); % Fit
line
x_fit = linspace(min(averageAutocraticLeadership),
max(averageAutocraticLeadership), 100);
```

```
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Autocratic Leadership');
ylabel('Average Task Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

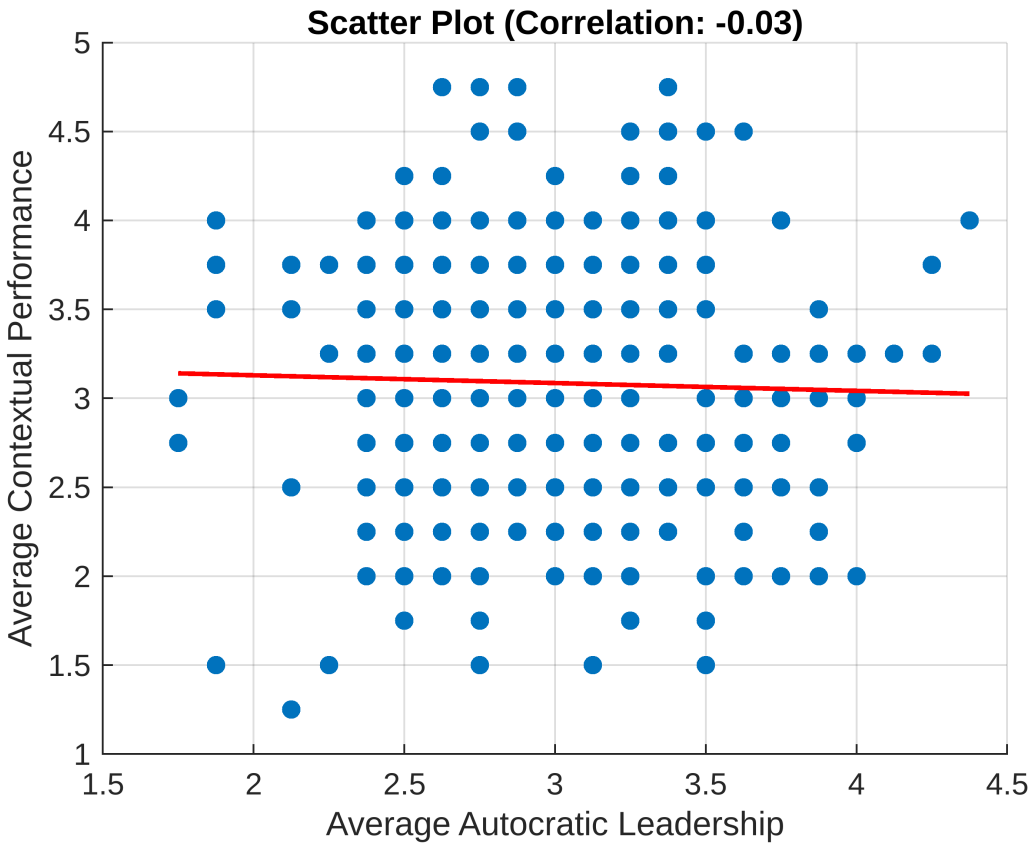



```
% Compute the correlation
correlationValue = corr(averageAutocraticLeadership,
averageContextualPerformance)
```

```
correlationValue =
-0.0296
```

```
% Scatter plot with trend line
scatter(averageAutocraticLeadership, averageContextualPerformance, 'filled');
hold on;
p = polyfit(averageAutocraticLeadership, averageContextualPerformance, 1); %
Fit line
x_fit = linspace(min(averageAutocraticLeadership),
max(averageAutocraticLeadership), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;
```

```
xlabel('Average Autocratic Leadership');
ylabel('Average Contextual Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```



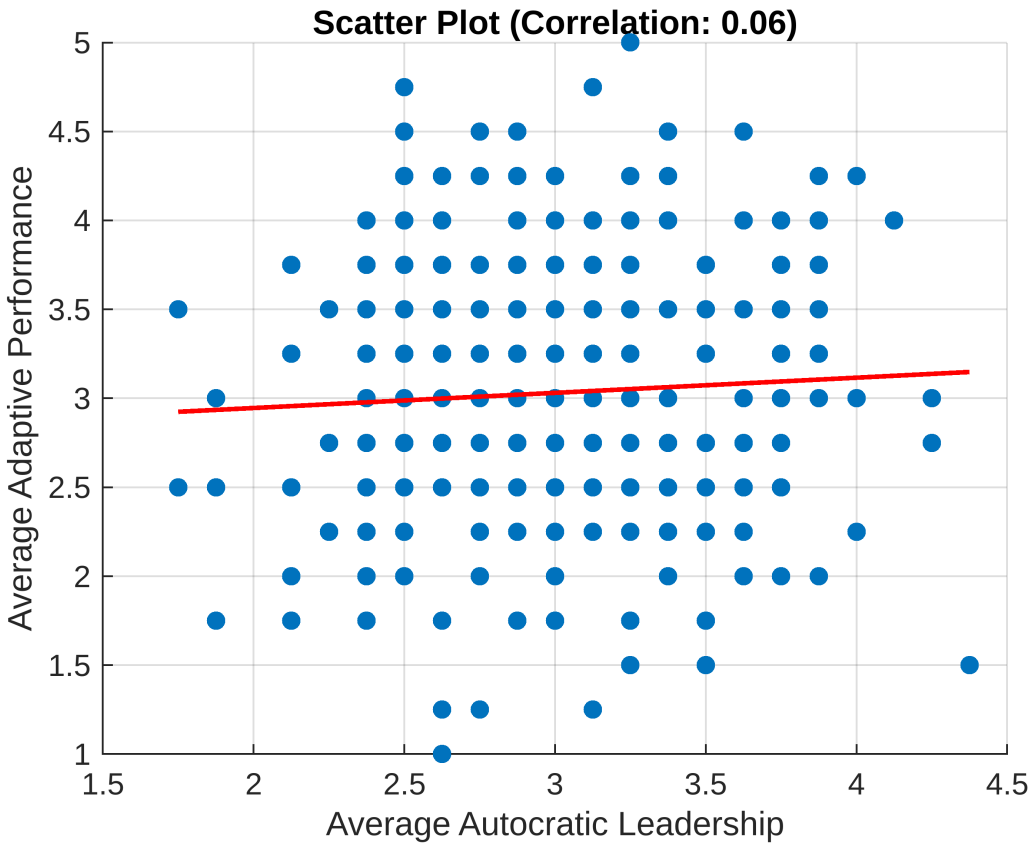

```
% Compute the correlation
correlationValue = corr(averageAutocraticLeadership,
averageAdaptivePerformance)
```

```
correlationValue =
0.0565
```

```
% Scatter plot with trend line
scatter(averageAutocraticLeadership, averageAdaptivePerformance, 'filled');
hold on;
p = polyfit(averageAutocraticLeadership, averageAdaptivePerformance, 1); %
Fit line
x_fit = linspace(min(averageAutocraticLeadership),
max(averageAutocraticLeadership), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Autocratic Leadership');
ylabel('Average Adaptive Performance');
```

```
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```



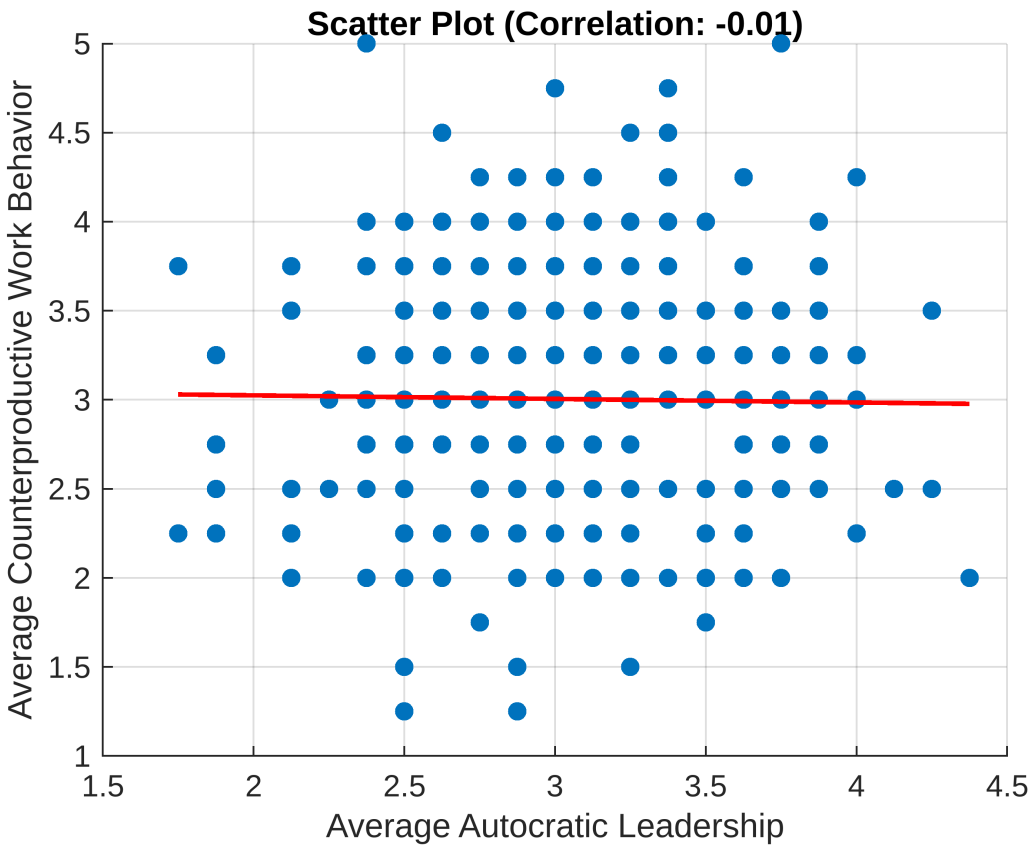

```
% Compute the correlation
correlationValue = corr(averageAutocraticLeadership,
averageCounterproductiveWorkBehavior)
```

```
correlationValue =
-0.0139
```

```
% Scatter plot with trend line
scatter(averageAutocraticLeadership, averageCounterproductiveWorkBehavior,
'filled');
hold on;
p = polyfit(averageAutocraticLeadership,
averageCounterproductiveWorkBehavior, 1); % Fit line
x_fit = linspace(min(averageAutocraticLeadership),
max(averageAutocraticLeadership), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Autocratic Leadership');
ylabel('Average Counterproductive Work Behavior');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
```

```
grid on;
```




```
% Combine all variables into a matrix
dataMatrix = [averageAutocraticLeadership,
averageCounterproductiveWorkBehavior, averageAdaptivePerformance,
averageContextualPerformance, averageTaskPerformance];

% Compute the correlation matrix
correlationMatrix = corr(dataMatrix);

% Display the correlation matrix
disp('Correlation Matrix:');
```

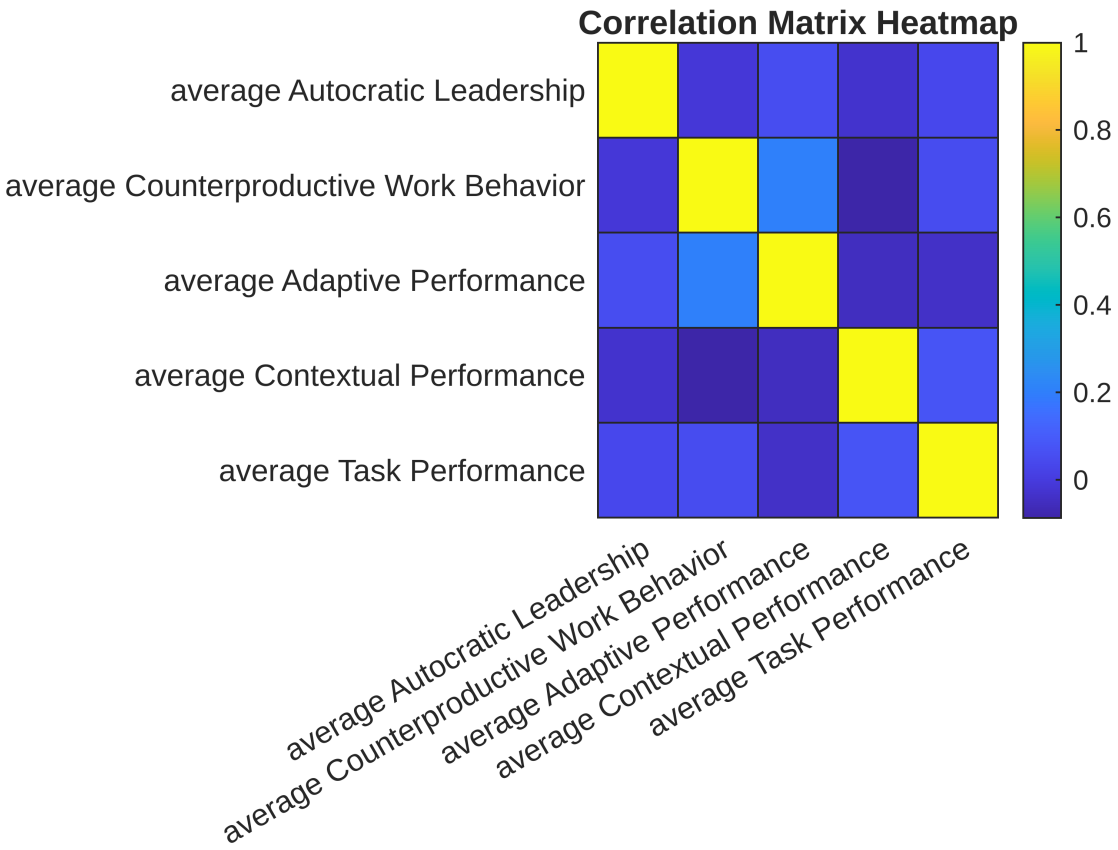
```
Correlation Matrix:
```

```
disp(correlationMatrix);
```

```
    1.0000   -0.0139    0.0565   -0.0296    0.0396
   -0.0139    1.0000    0.2026   -0.0877    0.0507
    0.0565    0.2026    1.0000   -0.0531   -0.0371
   -0.0296   -0.0877   -0.0531    1.0000    0.0763
    0.0396    0.0507   -0.0371    0.0763    1.0000
```

```
% Variable labels
variableNames = {'average Autocratic Leadership', 'average Counterproductive
Work Behavior', 'average Adaptive Performance', 'average Contextual
Performance', 'average Task Performance'};
```

```
% Create a heatmap
figure;
heatmap(variableNames, variableNames, correlationMatrix, 'Colormap', parula,
'ColorbarVisible', 'on');
title('Correlation Matrix Heatmap');
```



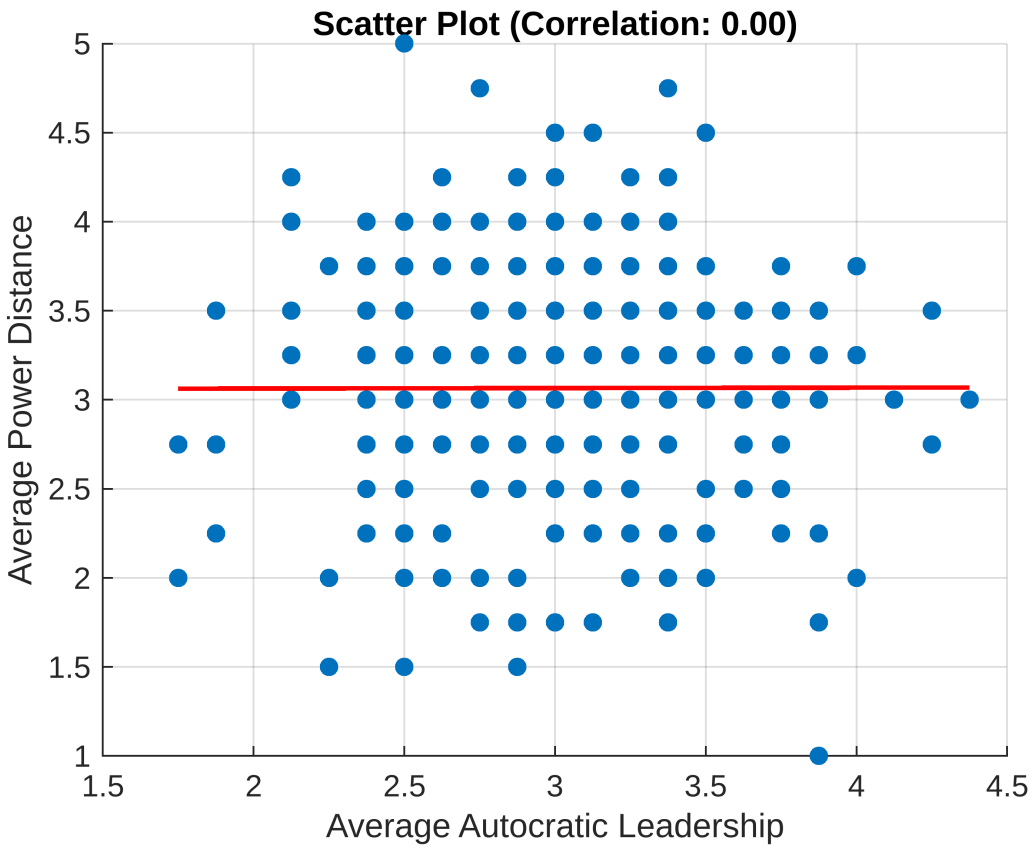

```
% Compute the correlation
correlationValue = corr(averageAutocraticLeadership, averagePower_Distance)
```

```
correlationValue =
0.0018
```

```
% Scatter plot with trend line
scatter(averageAutocraticLeadership, averagePower_Distance, 'filled');
hold on;
p = polyfit(averageAutocraticLeadership, averagePower_Distance, 1); % Fit
line
x_fit = linspace(min(averageAutocraticLeadership),
max(averageAutocraticLeadership), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Autocratic Leadership');
ylabel('Average Power Distance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
```

```
grid on;
```

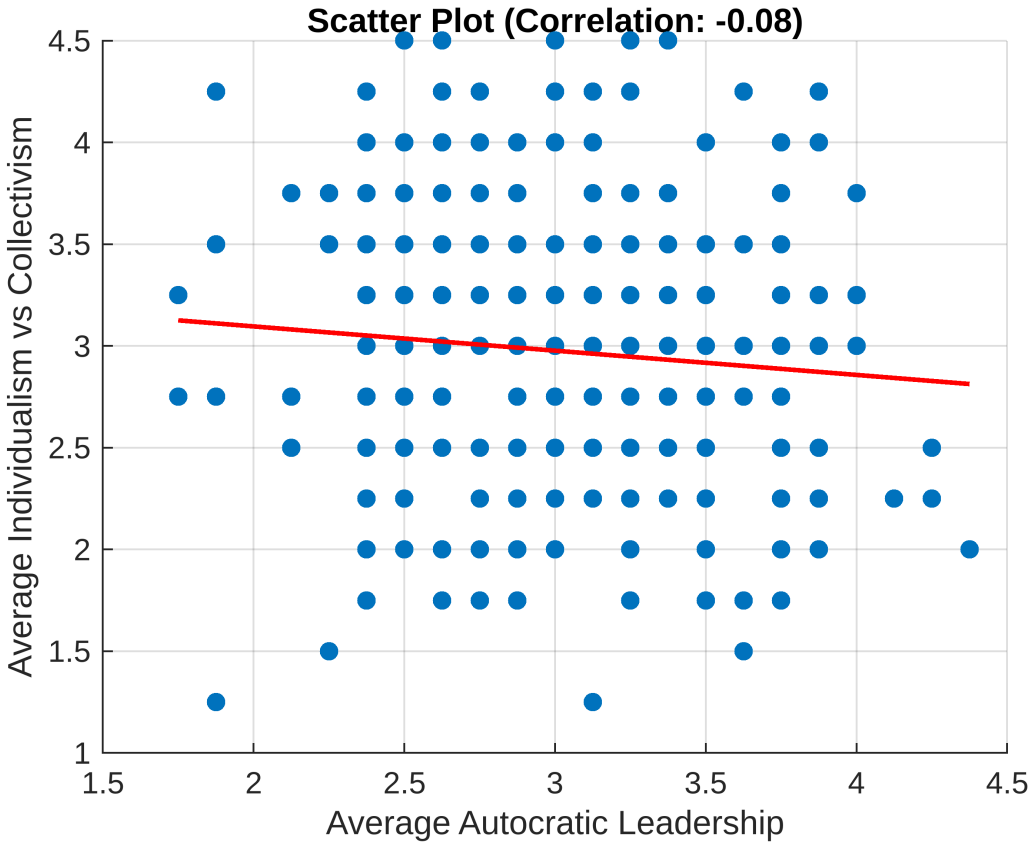



```
% Compute the correlation
correlationValue = corr(averageAutocraticLeadership,
averageIndividualismvsCollectivism)
```

```
correlationValue =
-0.0840
```

```
% Scatter plot with trend line
scatter(averageAutocraticLeadership, averageIndividualismvsCollectivism,
'filled');
hold on;
p = polyfit(averageAutocraticLeadership, averageIndividualismvsCollectivism,
1); % Fit line
x_fit = linspace(min(averageAutocraticLeadership),
max(averageAutocraticLeadership), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Autocratic Leadership');
ylabel('Average Individualism vs Collectivism');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

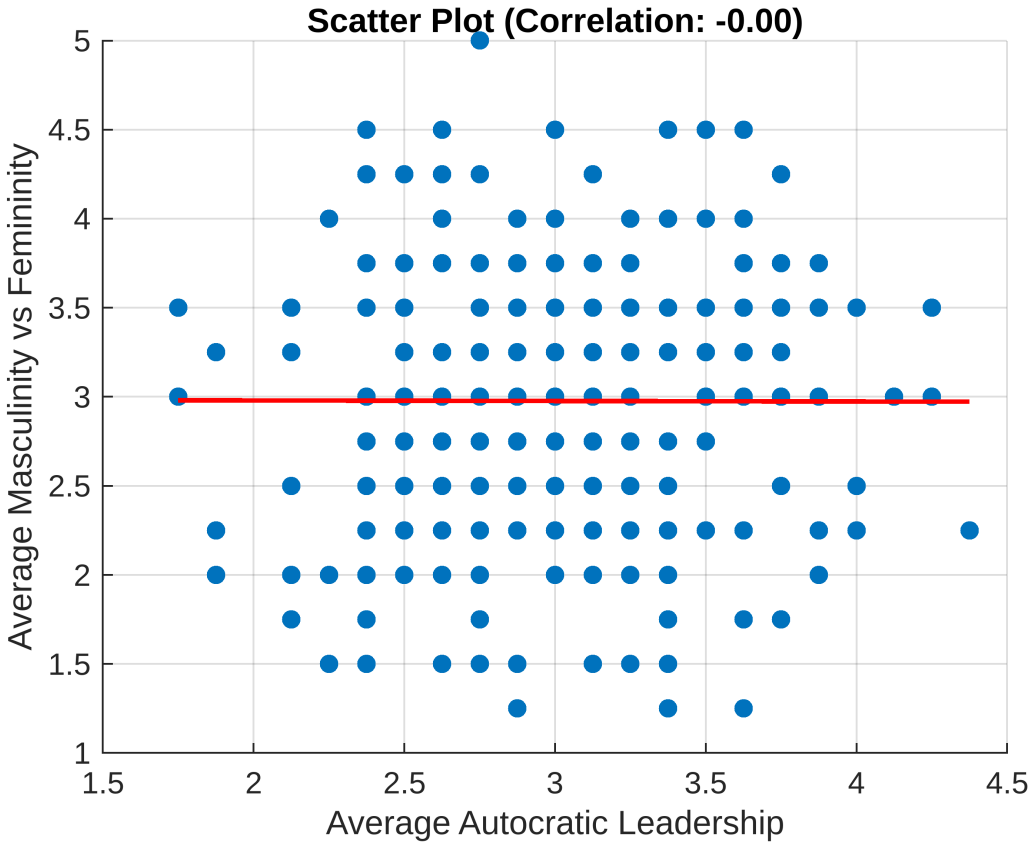



```
% Compute the correlation
correlationValue = corr(averageAutocraticLeadership,
averageMasculinityvsFemininity)
```

```
correlationValue =
-0.0020
```

```
% Scatter plot with trend line
scatter(averageAutocraticLeadership, averageMasculinityvsFemininity,
'filled');
hold on;
p = polyfit(averageAutocraticLeadership, averageMasculinityvsFemininity, 1);
% Fit line
x_fit = linspace(min(averageAutocraticLeadership),
max(averageAutocraticLeadership), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Autocratic Leadership');
ylabel('Average Masculinity vs Femininity');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

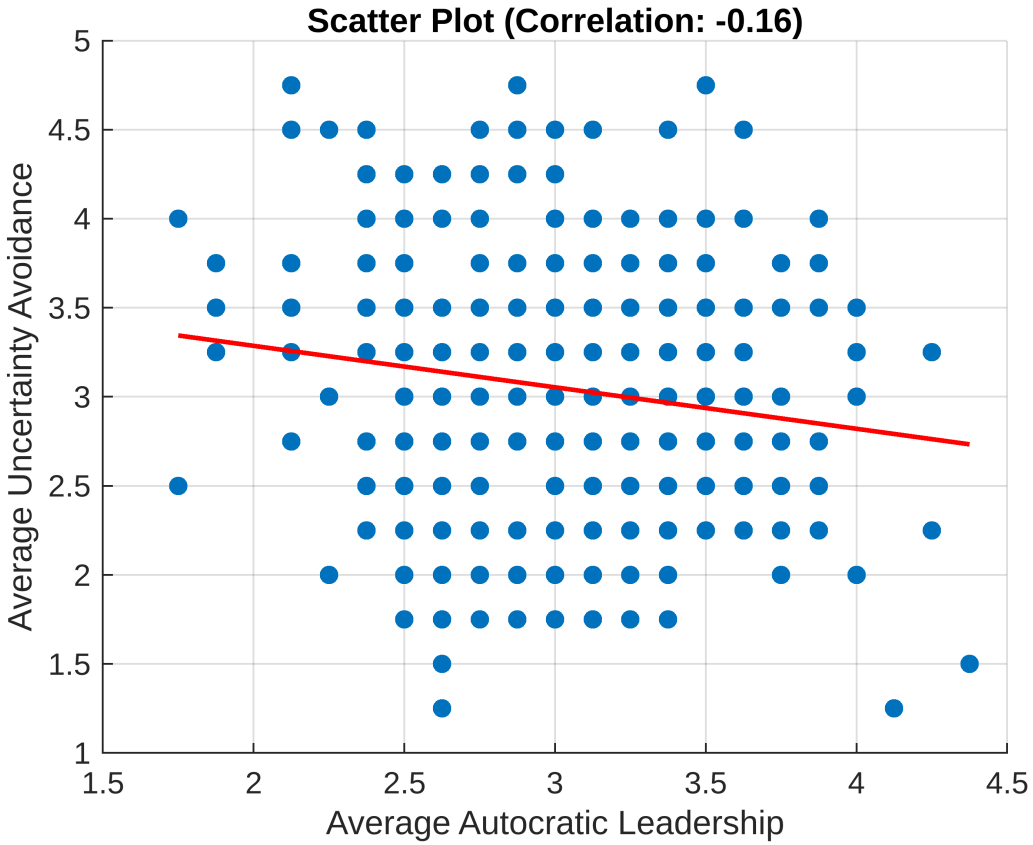



```
% Compute the correlation
correlationValue = corr(averageAutocraticLeadership,
averageUncertaintyAvoidance)
```

```
correlationValue =
-0.1572
```

```
% Scatter plot with trend line
scatter(averageAutocraticLeadership, averageUncertaintyAvoidance, 'filled');
hold on;
p = polyfit(averageAutocraticLeadership, averageUncertaintyAvoidance, 1); %
Fit line
x_fit = linspace(min(averageAutocraticLeadership),
max(averageAutocraticLeadership), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Autocratic Leadership');
ylabel('Average Uncertainty Avoidance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

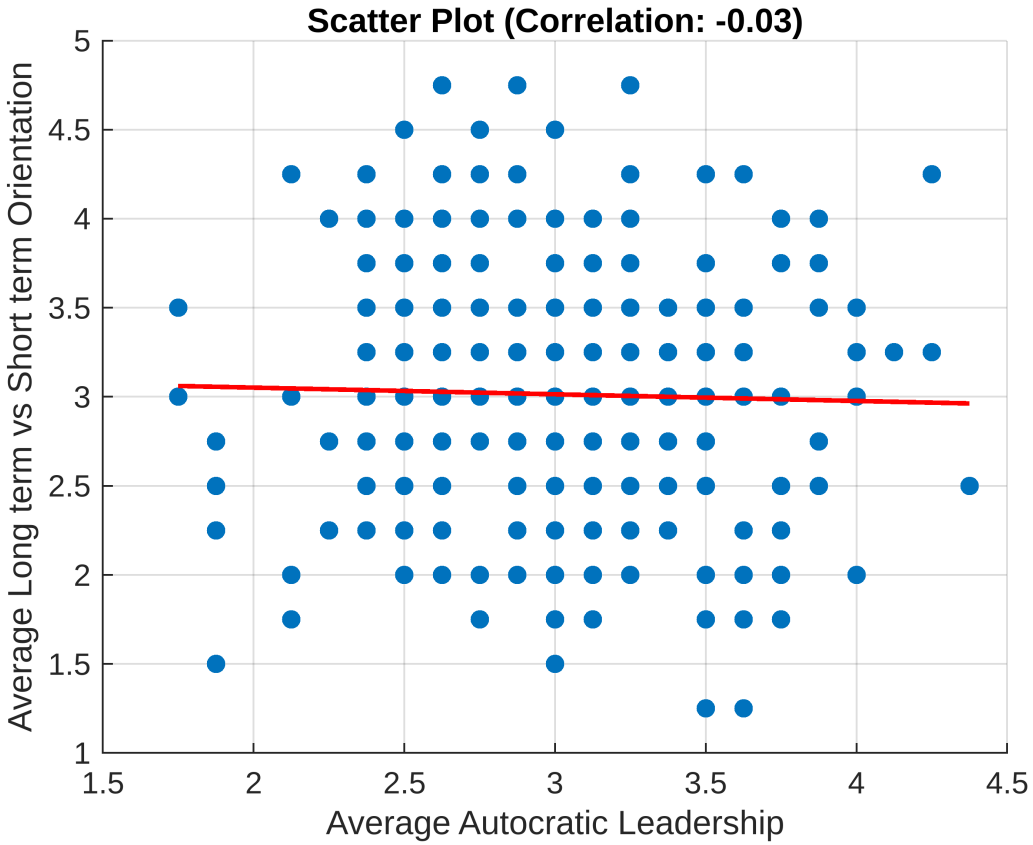



```
% Compute the correlation
correlationValue = corr(averageAutocraticLeadership,
averageLong_termvsShort_termOrientation)
```

```
correlationValue =
-0.0251
```

```
% Scatter plot with trend line
scatter(averageAutocraticLeadership,
averageLong_termvsShort_termOrientation, 'filled');
hold on;
p = polyfit(averageAutocraticLeadership,
averageLong_termvsShort_termOrientation, 1); % Fit line
x_fit = linspace(min(averageAutocraticLeadership),
max(averageAutocraticLeadership), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Autocratic Leadership');
ylabel('Average Long term vs Short term Orientation');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```




```
% Combine all variables into a matrix
dataMatrix = [averageAutocraticLeadership, averagePower_Distance,
averageIndividualismvsCollectivism, averageMasculinityvsFemininity,
averageUncertaintyAvoidance, averageLong_termvsShort_termOrientation];

% Compute the correlation matrix
correlationMatrix = corr(dataMatrix);

% Display the correlation matrix
disp('Correlation Matrix:');
```

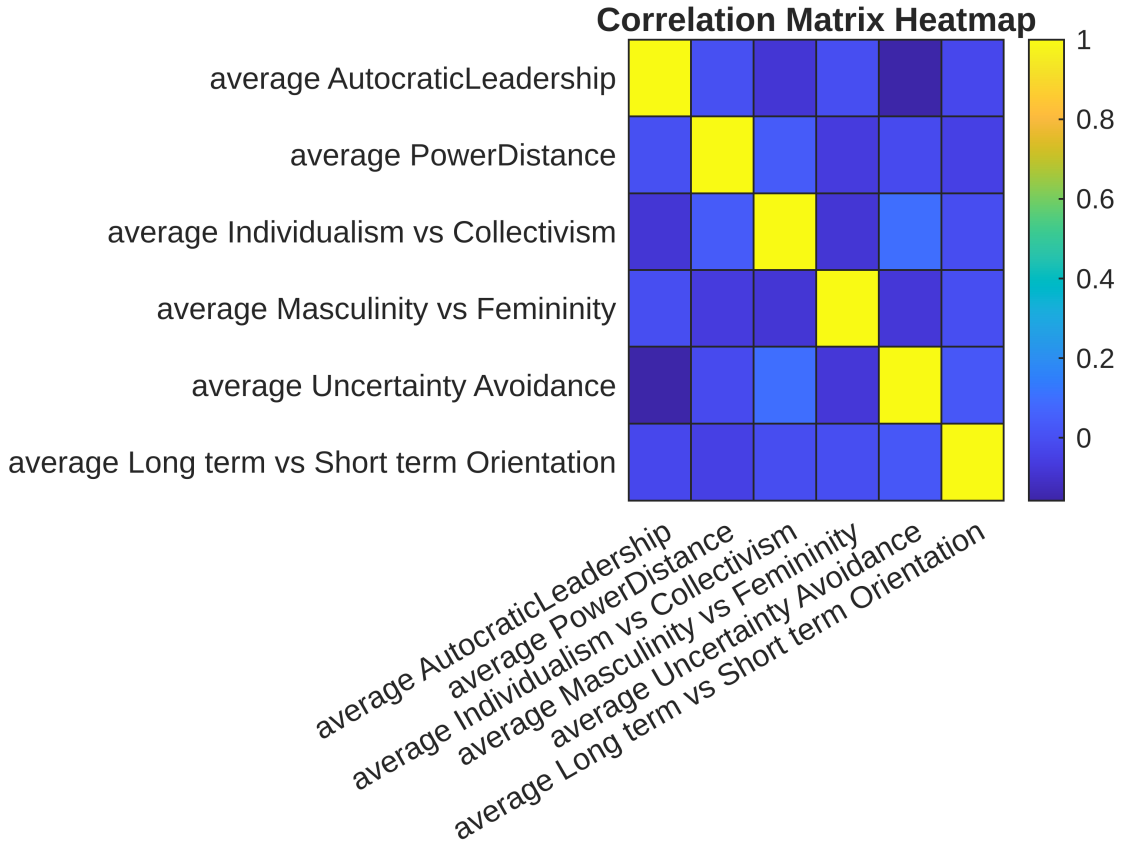
```
Correlation Matrix:
```

```
disp(correlationMatrix);
```

```
    1.0000    0.0018   -0.0840   -0.0020   -0.1572   -0.0251
    0.0018    1.0000    0.0413   -0.0625   -0.0169   -0.0473
   -0.0840    0.0413    1.0000   -0.0827    0.0977   -0.0055
   -0.0020   -0.0625   -0.0827    1.0000   -0.0798   -0.0002
   -0.1572   -0.0169    0.0977   -0.0798    1.0000    0.0261
   -0.0251   -0.0473   -0.0055   -0.0002    0.0261    1.0000
```

```
% Variable labels
variableNames = {'average AutocraticLeadership', 'average PowerDistance',
'average Individualism vs Collectivism', 'average Masculinity vs
Femininity', 'average Uncertainty Avoidance', 'average Long term vs Short
term Orientation'};
```

```
% Create a heatmap
figure;
heatmap(variableNames, variableNames, correlationMatrix, 'Colormap', parula,
'ColorbarVisible', 'on');
title('Correlation Matrix Heatmap');
```



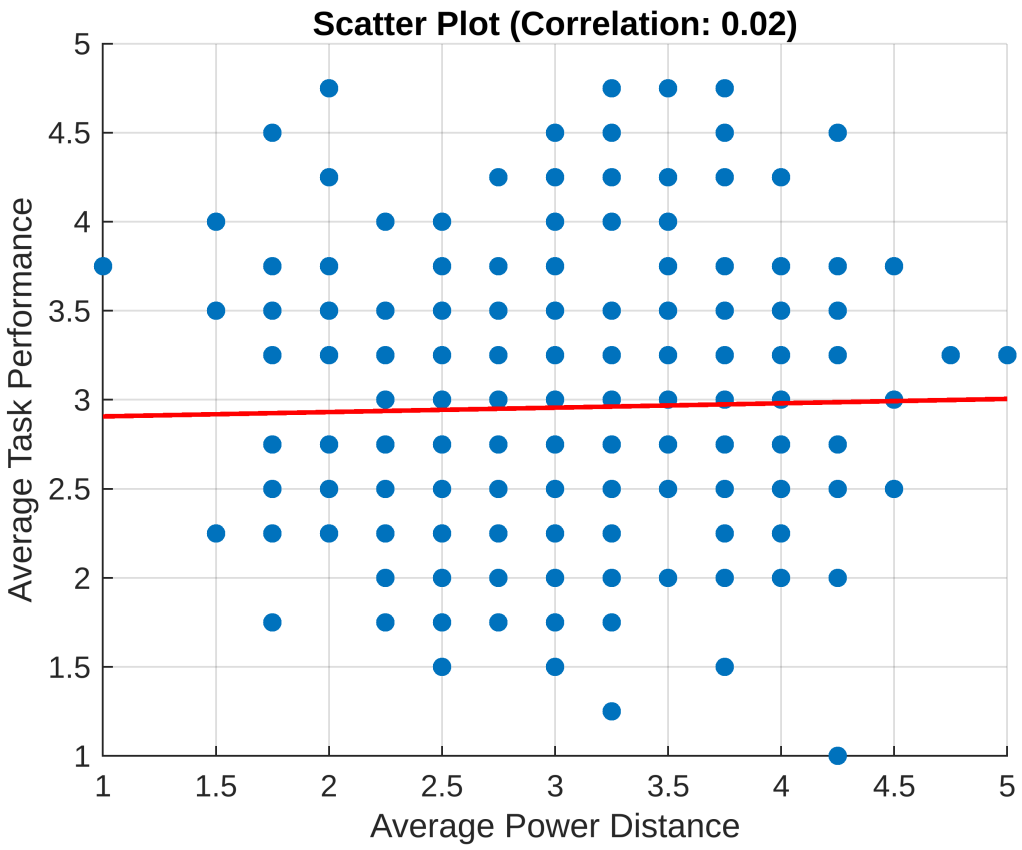

```
% Compute the correlation
correlationValue = corr(averagePower_Distance, averageTaskPerformance)
```

```
correlationValue =
0.0237
```

```
% Scatter plot with trend line
scatter(averagePower_Distance, averageTaskPerformance, 'filled');
hold on;
p = polyfit(averagePower_Distance, averageTaskPerformance, 1); % Fit line
x_fit = linspace(min(averagePower_Distance), max(averagePower_Distance),
100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Power Distance');
ylabel('Average Task Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
```

```
grid on;
```

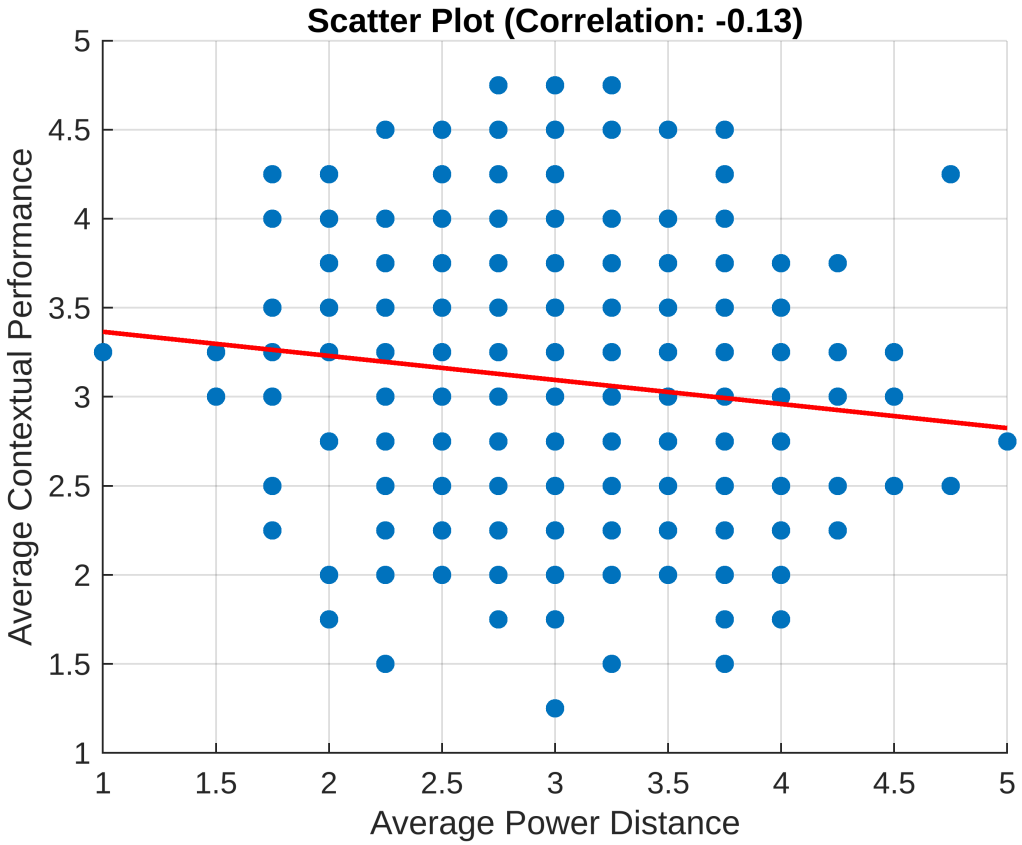



```
% Compute the correlation
correlationValue = corr(averagePower_Distance, averageContextualPerformance)
```

```
correlationValue =
-0.1272
```

```
% Scatter plot with trend line
scatter(averagePower_Distance, averageContextualPerformance, 'filled');
hold on;
p = polyfit(averagePower_Distance, averageContextualPerformance, 1); % Fit
line
x_fit = linspace(min(averagePower_Distance), max(averagePower_Distance),
100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Power Distance');
ylabel('Average Contextual Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

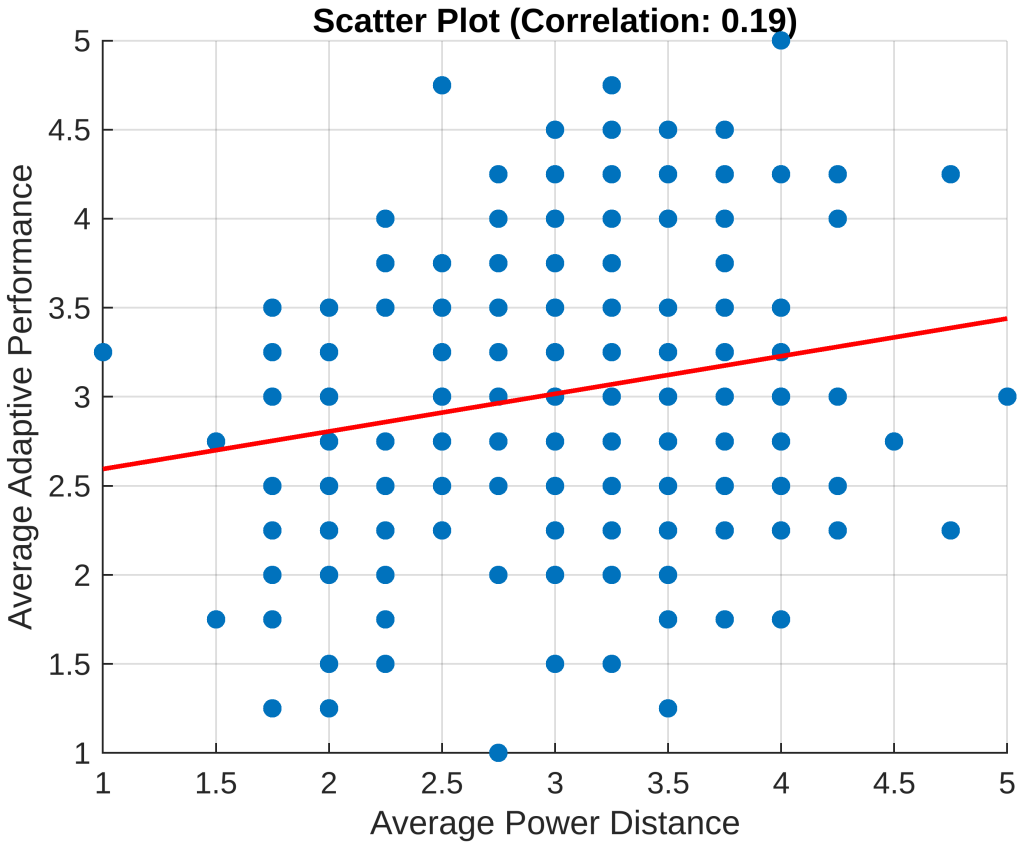



```
% Compute the correlation
correlationValue = corr(averagePower_Distance, averageAdaptivePerformance)
```

```
correlationValue =
0.1942
```

```
% Scatter plot with trend line
scatter(averagePower_Distance, averageAdaptivePerformance, 'filled');
hold on;
p = polyfit(averagePower_Distance, averageAdaptivePerformance, 1); % Fit line
x_fit = linspace(min(averagePower_Distance), max(averagePower_Distance),
100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Power Distance');
ylabel('Average Adaptive Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

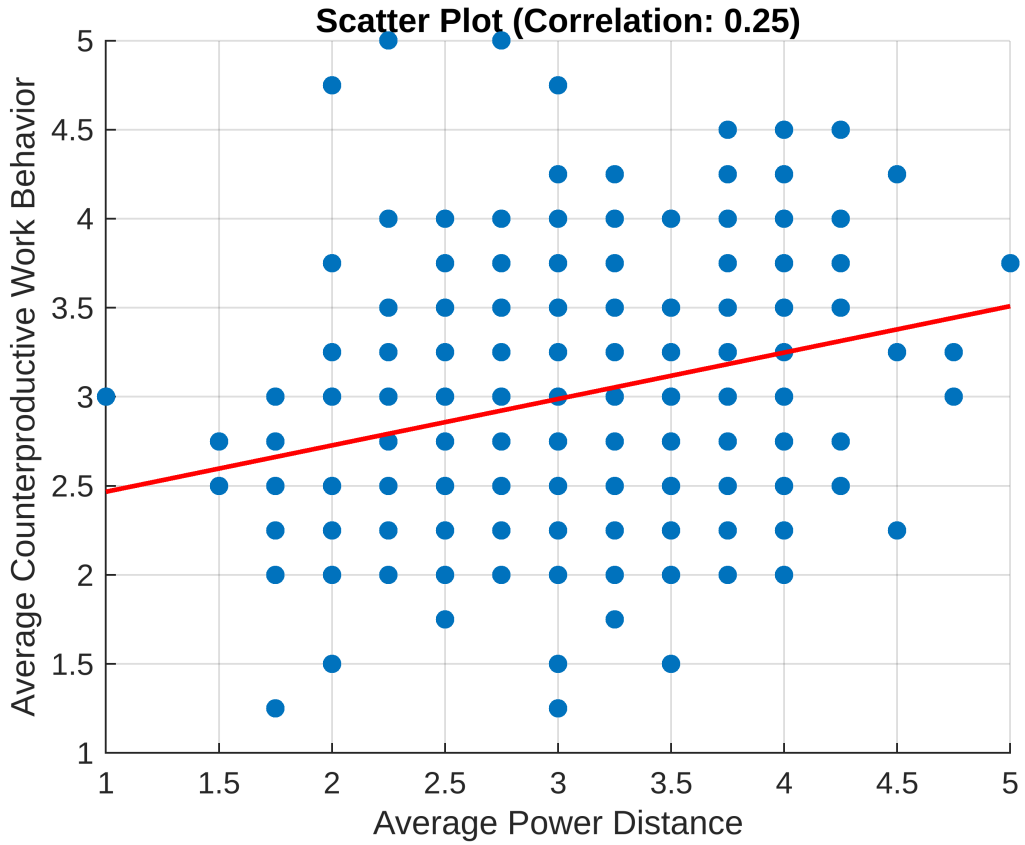



```
% Compute the correlation
correlationValue = corr(averagePower_Distance,
averageCounterproductiveWorkBehavior)
```

```
correlationValue =
0.2509
```

```
% Scatter plot with trend line
scatter(averagePower_Distance, averageCounterproductiveWorkBehavior,
'filled');
hold on;
p = polyfit(averagePower_Distance, averageCounterproductiveWorkBehavior, 1);
% Fit line
x_fit = linspace(min(averagePower_Distance), max(averagePower_Distance),
100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Power Distance');
ylabel('Average Counterproductive Work Behavior');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```




```
% Combine all variables into a matrix
dataMatrix = [averagePower_Distance, averageTaskPerformance,
averageContextualPerformance, averageAdaptivePerformance,
averageCounterproductiveWorkBehavior];

% Compute the correlation matrix
correlationMatrix = corr(dataMatrix);

% Display the correlation matrix
disp('Correlation Matrix:');
```

```
Correlation Matrix:
```

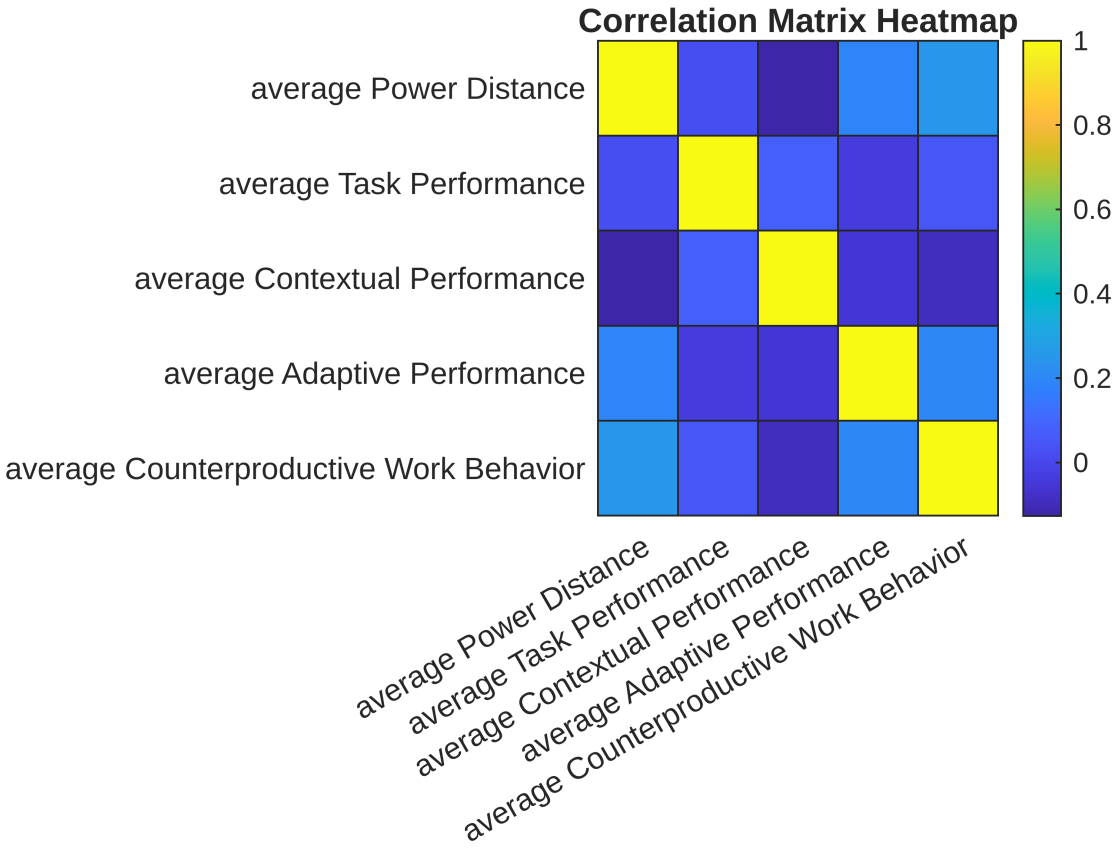
```
disp(correlationMatrix);
```

```
    1.0000    0.0237   -0.1272    0.1942    0.2509
    0.0237    1.0000    0.0763   -0.0371    0.0507
   -0.1272    0.0763    1.0000   -0.0531   -0.0877
    0.1942   -0.0371   -0.0531    1.0000    0.2026
    0.2509    0.0507   -0.0877    0.2026    1.0000
```

```
% Variable labels
variableNames = {'average Power Distance', 'average Task Performance',
'average Contextual Performance', 'average Adaptive Performance', 'average
Counterproductive Work Behavior'};

% Create a heatmap
```

```
figure;
heatmap(variableNames, variableNames, correlationMatrix, 'Colormap', parula,
'ColorbarVisible', 'on');
title('Correlation Matrix Heatmap');
```



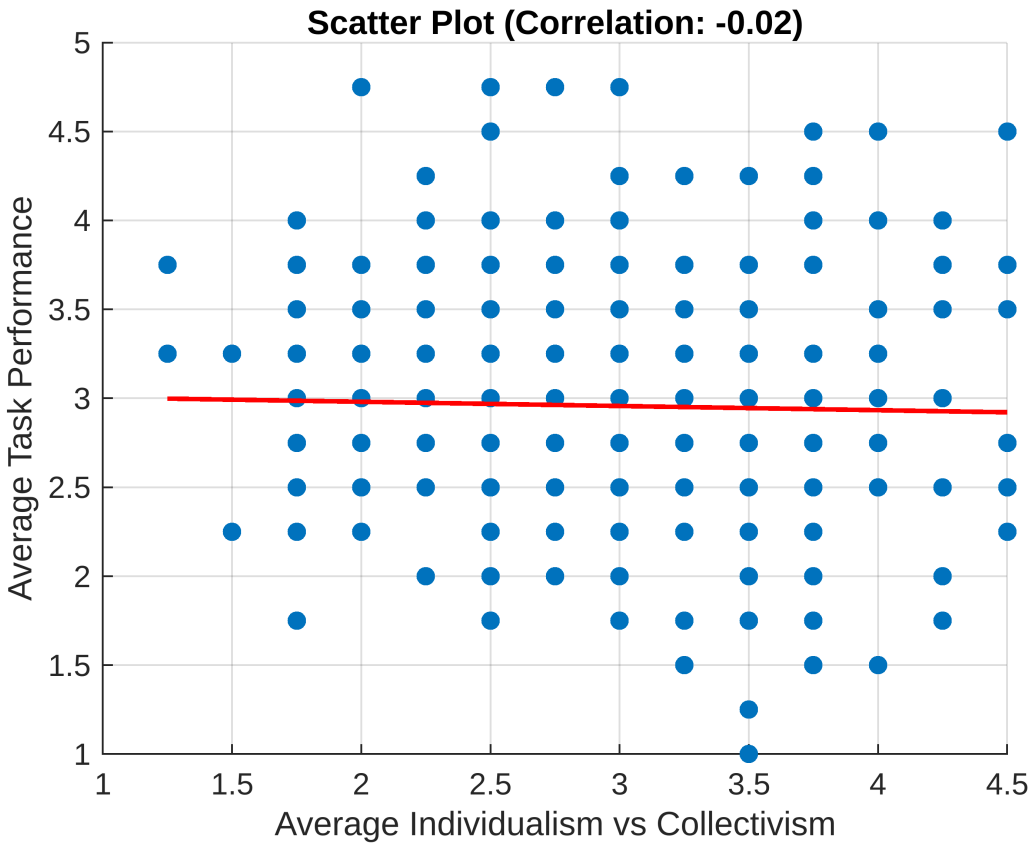

```
% Compute the correlation
correlationValue = corr(averageIndividualismvsCollectivism,
averageTaskPerformance)
```

```
correlationValue =
-0.0233
```

```
% Scatter plot with trend line
scatter(averageIndividualismvsCollectivism, averageTaskPerformance,
'filled');
hold on;
p = polyfit(averageIndividualismvsCollectivism, averageTaskPerformance, 1);
% Fit line
x_fit = linspace(min(averageIndividualismvsCollectivism),
max(averageIndividualismvsCollectivism), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Individualism vs Collectivism');
ylabel('Average Task Performance');
```

```
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```



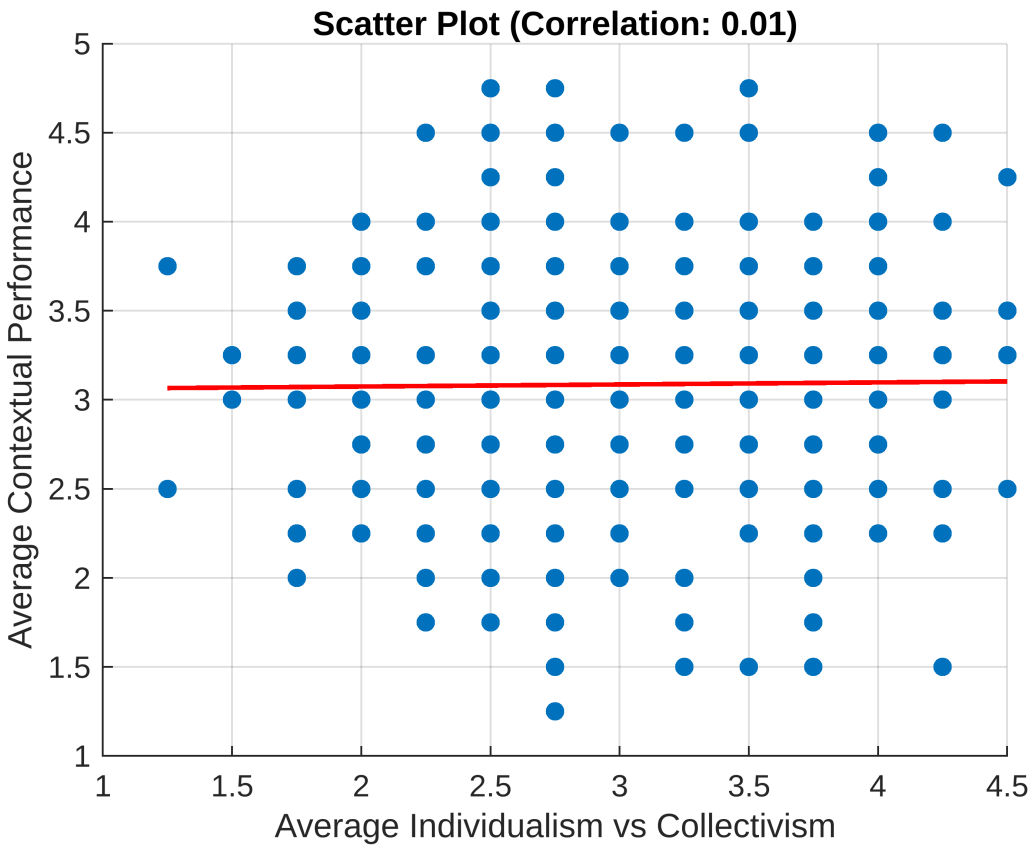

```
% Compute the correlation
correlationValue = corr(averageIndividualismvsCollectivism,
averageContextualPerformance)
```

```
correlationValue =
0.0111
```

```
% Scatter plot with trend line
scatter(averageIndividualismvsCollectivism, averageContextualPerformance,
'filled');
hold on;
p = polyfit(averageIndividualismvsCollectivism,
averageContextualPerformance, 1); % Fit line
x_fit = linspace(min(averageIndividualismvsCollectivism),
max(averageIndividualismvsCollectivism), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Individualism vs Collectivism');
ylabel('Average Contextual Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
```

```
grid on;
```

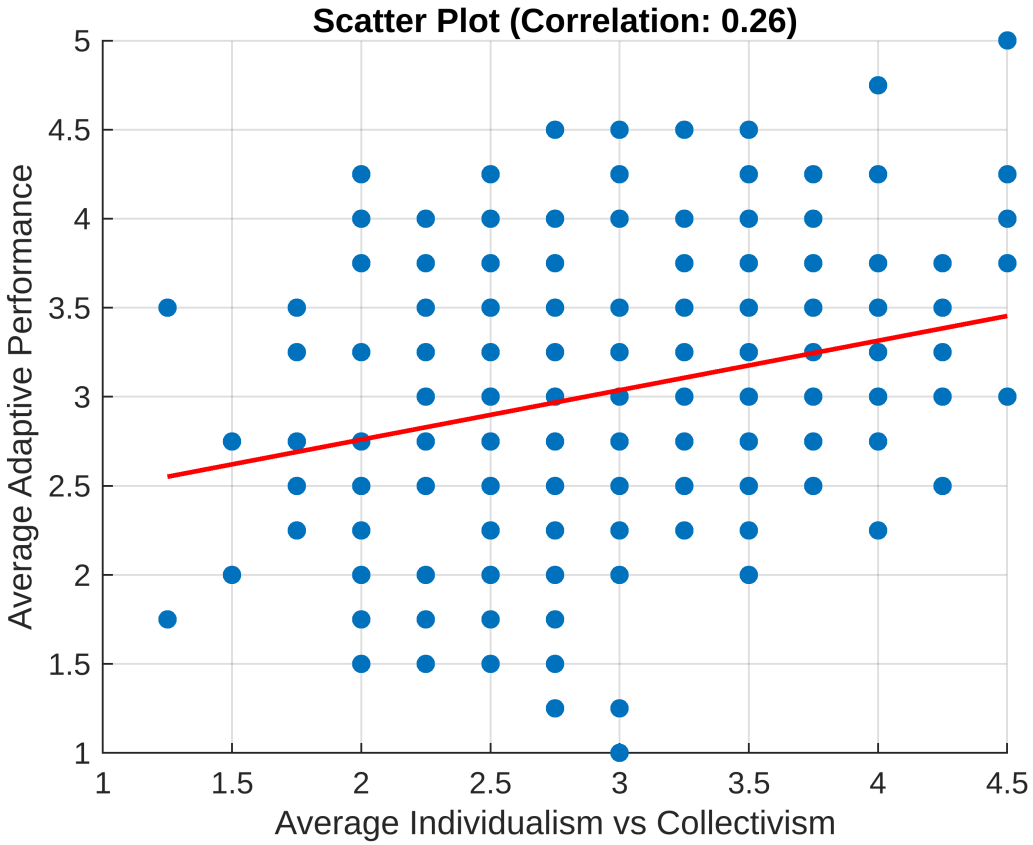



```
% Compute the correlation
correlationValue = corr(averageIndividualismvsCollectivism,
averageAdaptivePerformance)
```

```
correlationValue =
0.2611
```

```
% Scatter plot with trend line
scatter(averageIndividualismvsCollectivism, averageAdaptivePerformance,
'filled');
hold on;
p = polyfit(averageIndividualismvsCollectivism, averageAdaptivePerformance,
1); % Fit line
x_fit = linspace(min(averageIndividualismvsCollectivism),
max(averageIndividualismvsCollectivism), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Individualism vs Collectivism');
ylabel('Average Adaptive Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

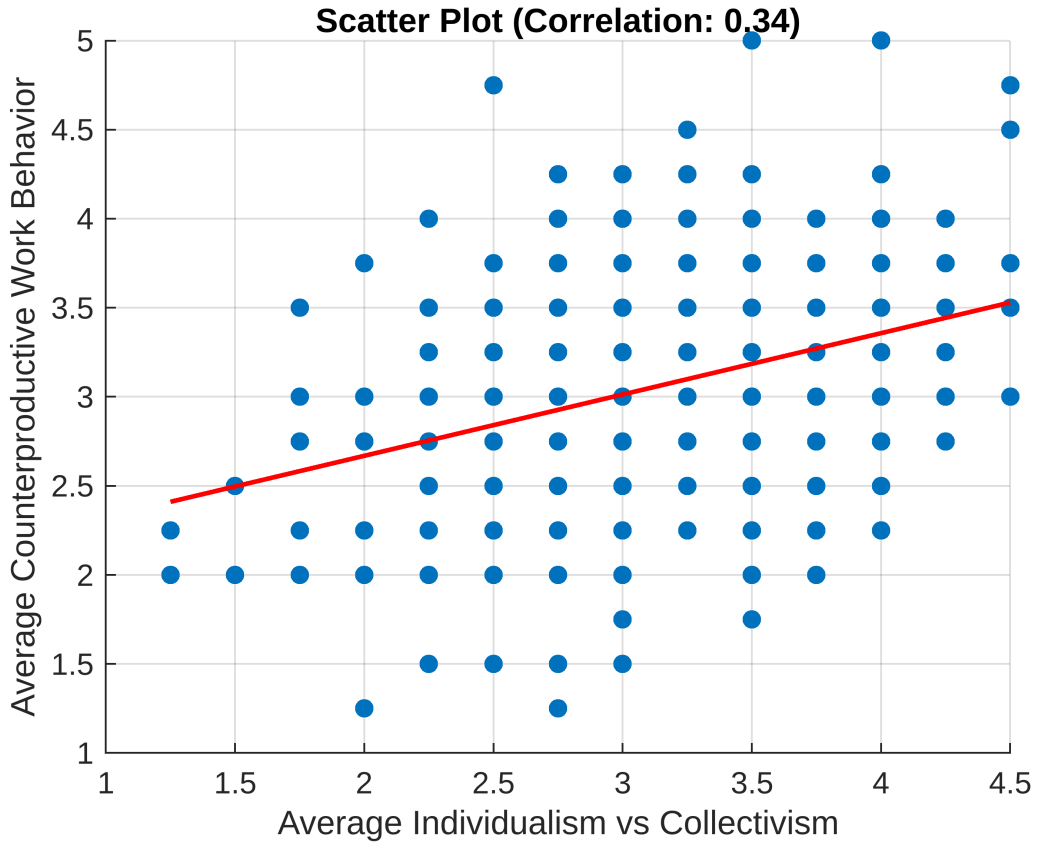



```
% Compute the correlation
correlationValue = corr(averageIndividualismvsCollectivism,
averageCounterproductiveWorkBehavior)
```

```
correlationValue =
0.3391
```

```
% Scatter plot with trend line
scatter(averageIndividualismvsCollectivism,
averageCounterproductiveWorkBehavior, 'filled');
hold on;
p = polyfit(averageIndividualismvsCollectivism,
averageCounterproductiveWorkBehavior, 1); % Fit line
x_fit = linspace(min(averageIndividualismvsCollectivism),
max(averageIndividualismvsCollectivism), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Individualism vs Collectivism');
ylabel('Average Counterproductive Work Behavior');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```




```
% Combine all variables into a matrix
dataMatrix = [averageIndividualismvsCollectivism, averageTaskPerformance,
averageContextualPerformance, averageAdaptivePerformance,
averageCounterproductiveWorkBehavior];

% Compute the correlation matrix
correlationMatrix = corr(dataMatrix);

% Display the correlation matrix
disp('Correlation Matrix:');
```

```
Correlation Matrix:
```

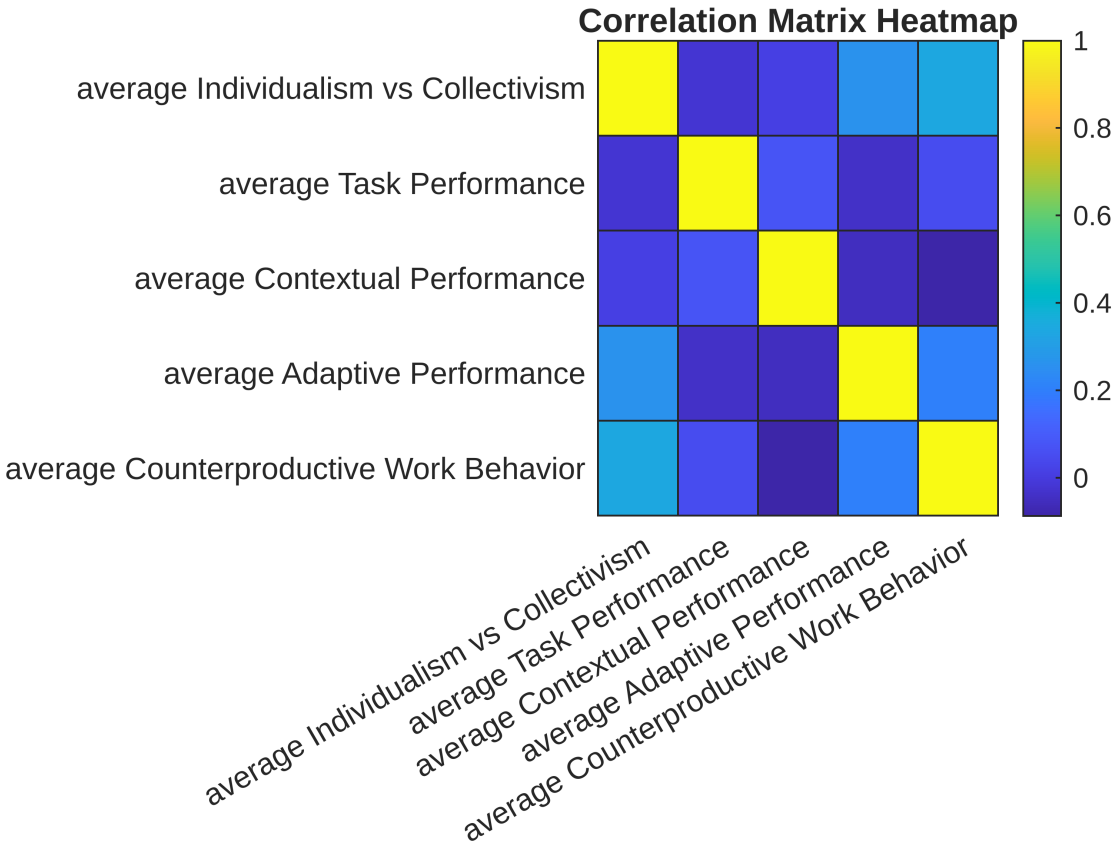
```
disp(correlationMatrix);
```

```
    1.0000   -0.0233    0.0111    0.2611    0.3391
   -0.0233    1.0000    0.0763   -0.0371    0.0507
    0.0111    0.0763    1.0000   -0.0531   -0.0877
    0.2611   -0.0371   -0.0531    1.0000    0.2026
    0.3391    0.0507   -0.0877    0.2026    1.0000
```

```
% Variable labels
variableNames = {'average Individualism vs Collectivism', 'average
Task Performance', 'average Contextual Performance', 'average Adaptive
Performance', 'average Counterproductive Work Behavior'};

% Create a heatmap
```

```
figure;
heatmap(variableNames, variableNames, correlationMatrix, 'Colormap', parula,
'ColorbarVisible', 'on');
title('Correlation Matrix Heatmap');
```



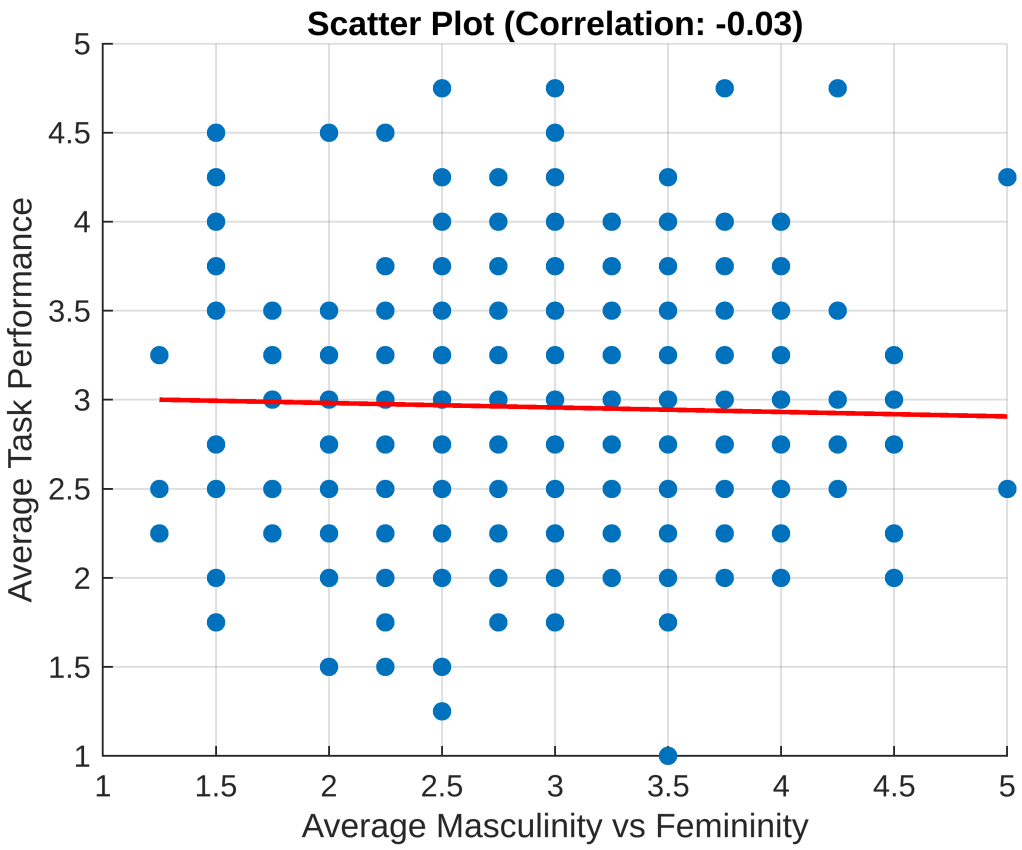

```
% Compute the correlation
correlationValue = corr(averageMasculinityvsFemininity,
averageTaskPerformance)
```

```
correlationValue =
-0.0267
```

```
% Scatter plot with trend line
scatter(averageMasculinityvsFemininity, averageTaskPerformance, 'filled');
hold on;
p = polyfit(averageMasculinityvsFemininity, averageTaskPerformance, 1); %
Fit line
x_fit = linspace(min(averageMasculinityvsFemininity),
max(averageMasculinityvsFemininity), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Masculinity vs Femininity');
ylabel('Average Task Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
```

```
grid on;
```

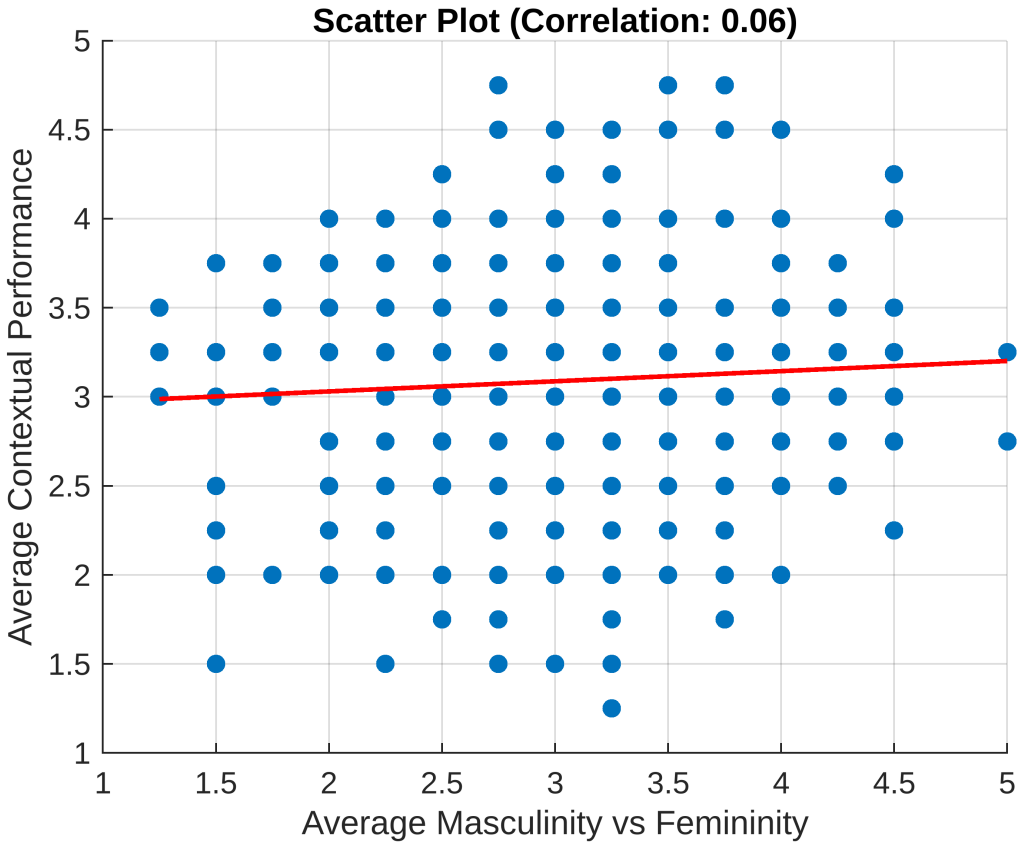



```
% Compute the correlation
correlationValue = corr(averageMasculinityvsFemininity,
averageContextualPerformance)
```

```
correlationValue =
0.0592
```

```
% Scatter plot with trend line
scatter(averageMasculinityvsFemininity, averageContextualPerformance,
'filled');
hold on;
p = polyfit(averageMasculinityvsFemininity, averageContextualPerformance,
1); % Fit line
x_fit = linspace(min(averageMasculinityvsFemininity),
max(averageMasculinityvsFemininity), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Masculinity vs Femininity');
ylabel('Average Contextual Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

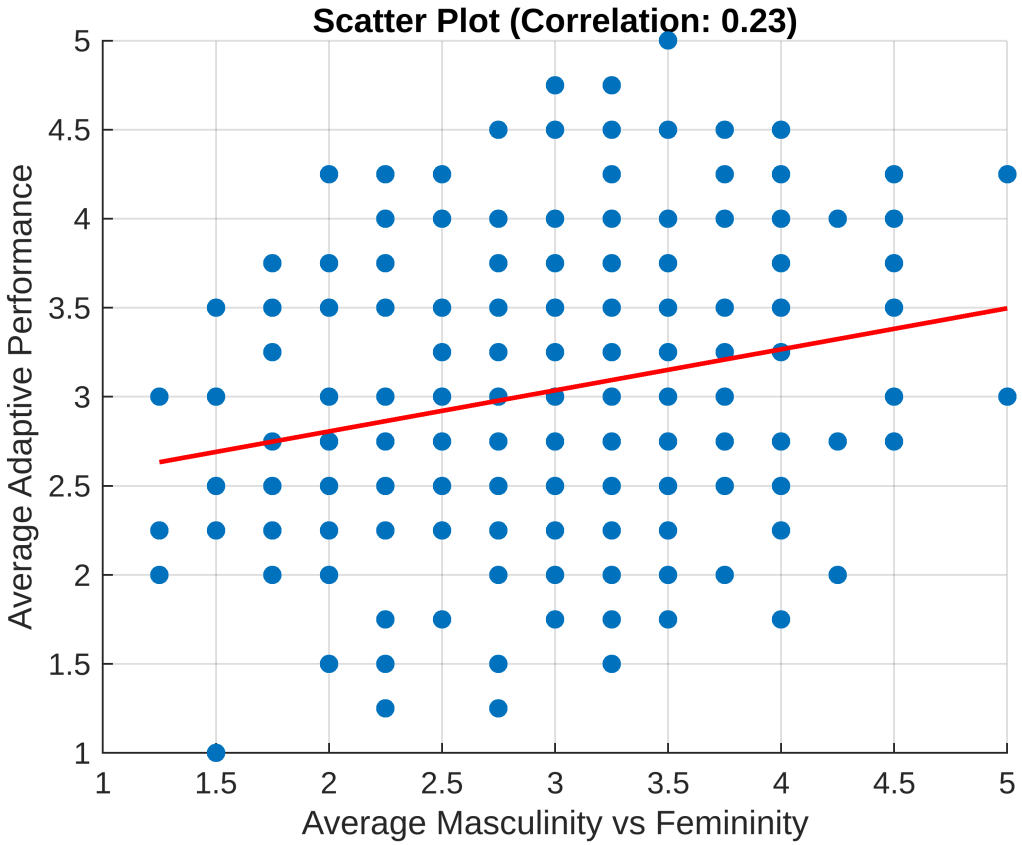



```
% Compute the correlation
correlationValue = corr(averageMasculinityvsFemininity,
averageAdaptivePerformance)
```

```
correlationValue =
0.2339
```

```
% Scatter plot with trend line
scatter(averageMasculinityvsFemininity, averageAdaptivePerformance,
'filled');
hold on;
p = polyfit(averageMasculinityvsFemininity, averageAdaptivePerformance, 1);
% Fit line
x_fit = linspace(min(averageMasculinityvsFemininity),
max(averageMasculinityvsFemininity), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Masculinity vs Femininity');
ylabel('Average Adaptive Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

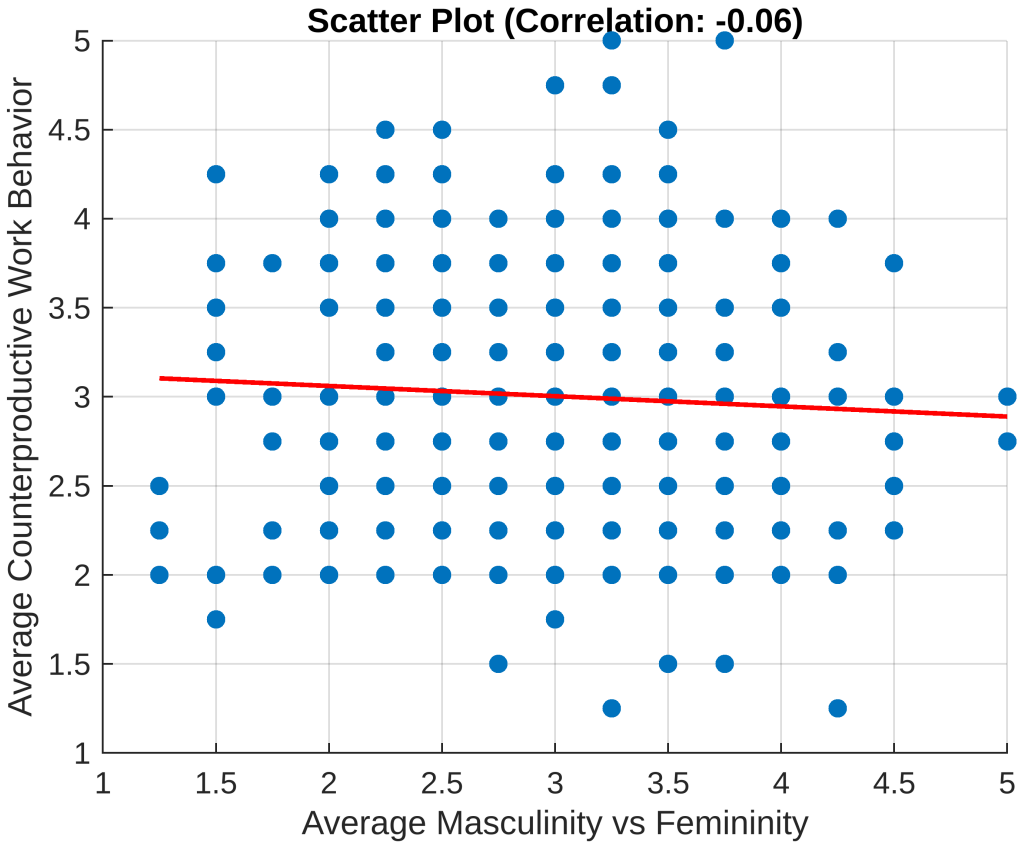



```
% Compute the correlation
correlationValue = corr(averageMasculinityvsFemininity,
averageCounterproductiveWorkBehavior)
```

```
correlationValue =
-0.0609
```

```
% Scatter plot with trend line
scatter(averageMasculinityvsFemininity,
averageCounterproductiveWorkBehavior, 'filled');
hold on;
p = polyfit(averageMasculinityvsFemininity,
averageCounterproductiveWorkBehavior, 1); % Fit line
x_fit = linspace(min(averageMasculinityvsFemininity),
max(averageMasculinityvsFemininity), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Masculinity vs Femininity');
ylabel('Average Counterproductive Work Behavior');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

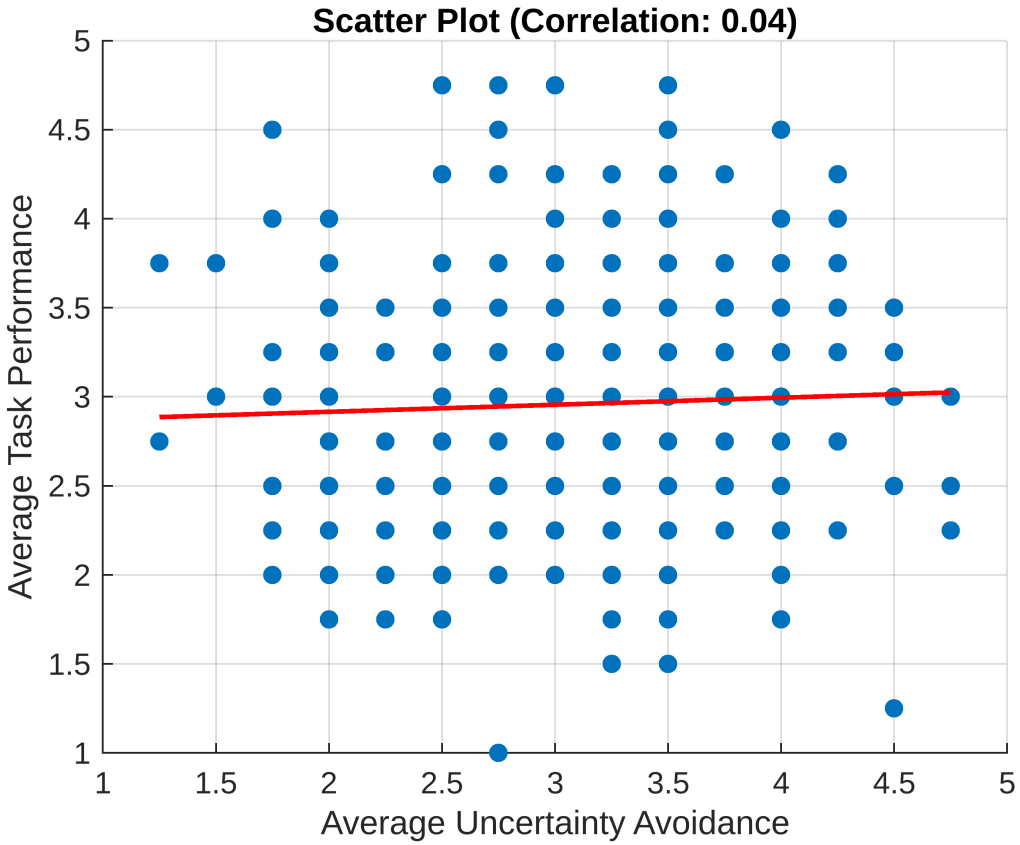



```
% Compute the correlation
correlationValue = corr(averageUncertaintyAvoidance, averageTaskPerformance)
```

```
correlationValue =
0.0407
```

```
% Scatter plot with trend line
scatter(averageUncertaintyAvoidance, averageTaskPerformance, 'filled');
hold on;
p = polyfit(averageUncertaintyAvoidance, averageTaskPerformance, 1); % Fit
line
x_fit = linspace(min(averageUncertaintyAvoidance),
max(averageUncertaintyAvoidance), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Uncertainty Avoidance');
ylabel('Average Task Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

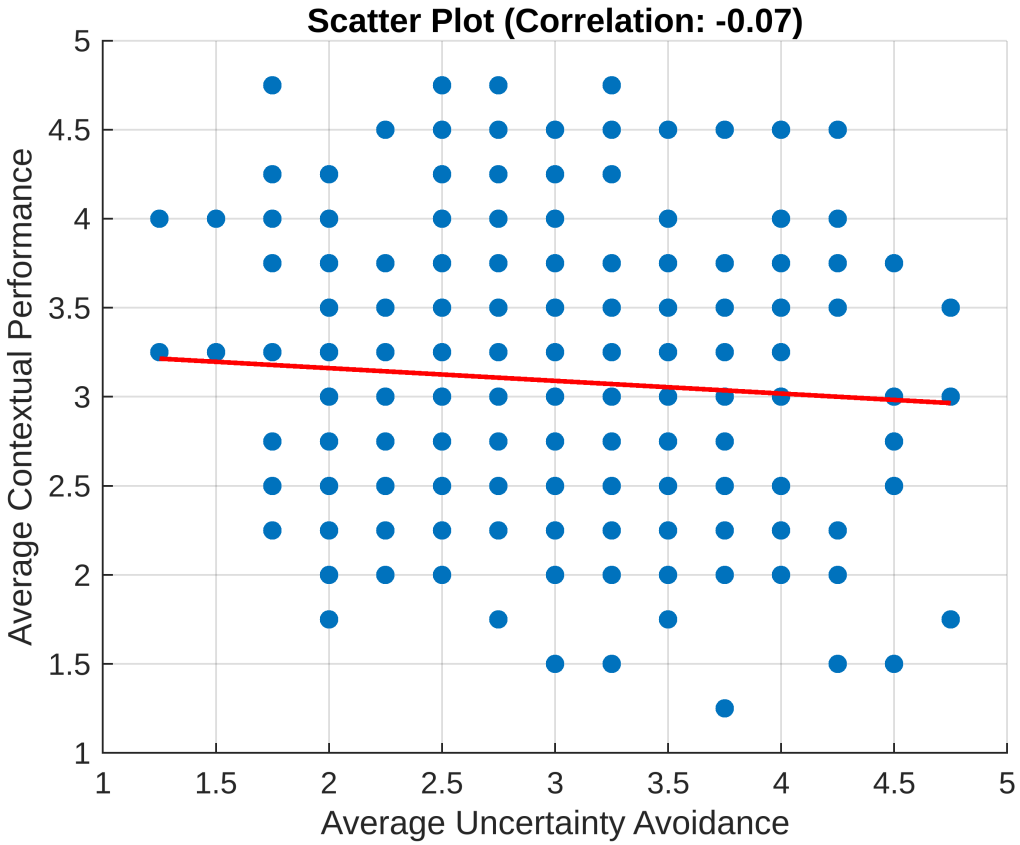



```
% Compute the correlation
correlationValue = corr(averageUncertaintyAvoidance,
averageContextualPerformance)
```

```
correlationValue =
-0.0717
```

```
% Scatter plot with trend line
scatter(averageUncertaintyAvoidance, averageContextualPerformance, 'filled');
hold on;
p = polyfit(averageUncertaintyAvoidance, averageContextualPerformance, 1); %
Fit line
x_fit = linspace(min(averageUncertaintyAvoidance),
max(averageUncertaintyAvoidance), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Uncertainty Avoidance');
ylabel('Average Contextual Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

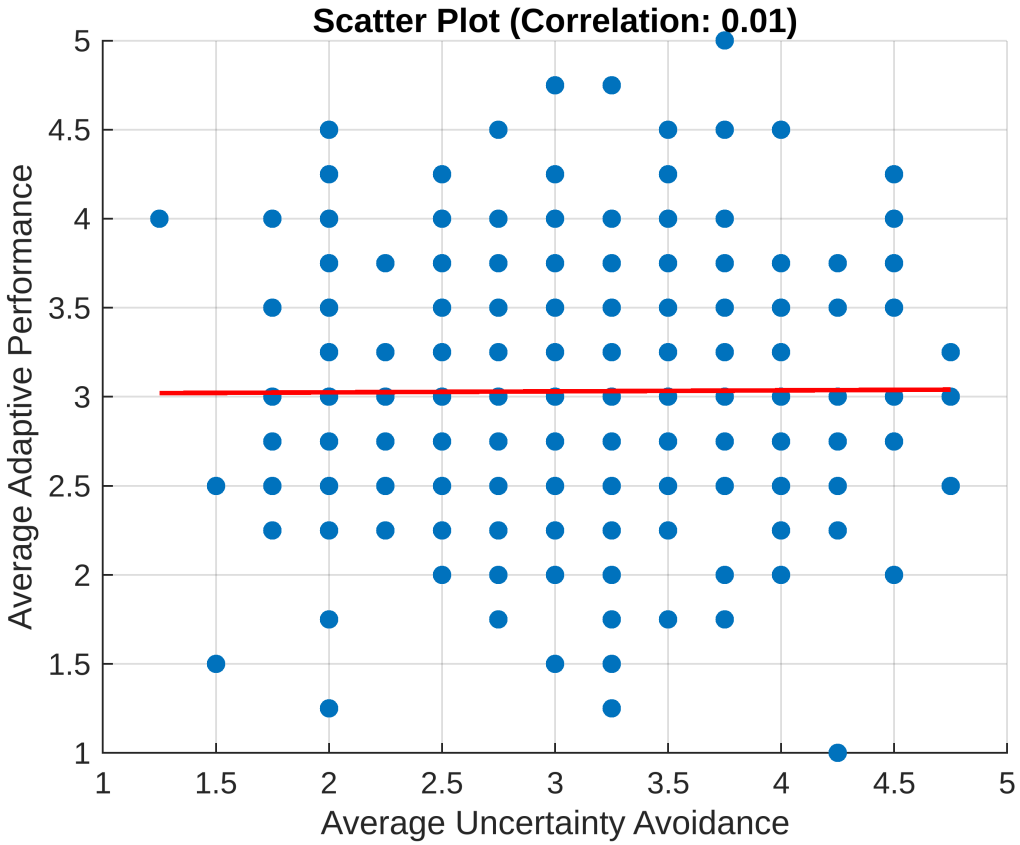



```
% Compute the correlation
correlationValue = corr(averageUncertaintyAvoidance,
averageAdaptivePerformance)
```

```
correlationValue =
0.0055
```

```
% Scatter plot with trend line
scatter(averageUncertaintyAvoidance, averageAdaptivePerformance, 'filled');
hold on;
p = polyfit(averageUncertaintyAvoidance, averageAdaptivePerformance, 1); %
Fit line
x_fit = linspace(min(averageUncertaintyAvoidance),
max(averageUncertaintyAvoidance), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Uncertainty Avoidance');
ylabel('Average Adaptive Performance');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```

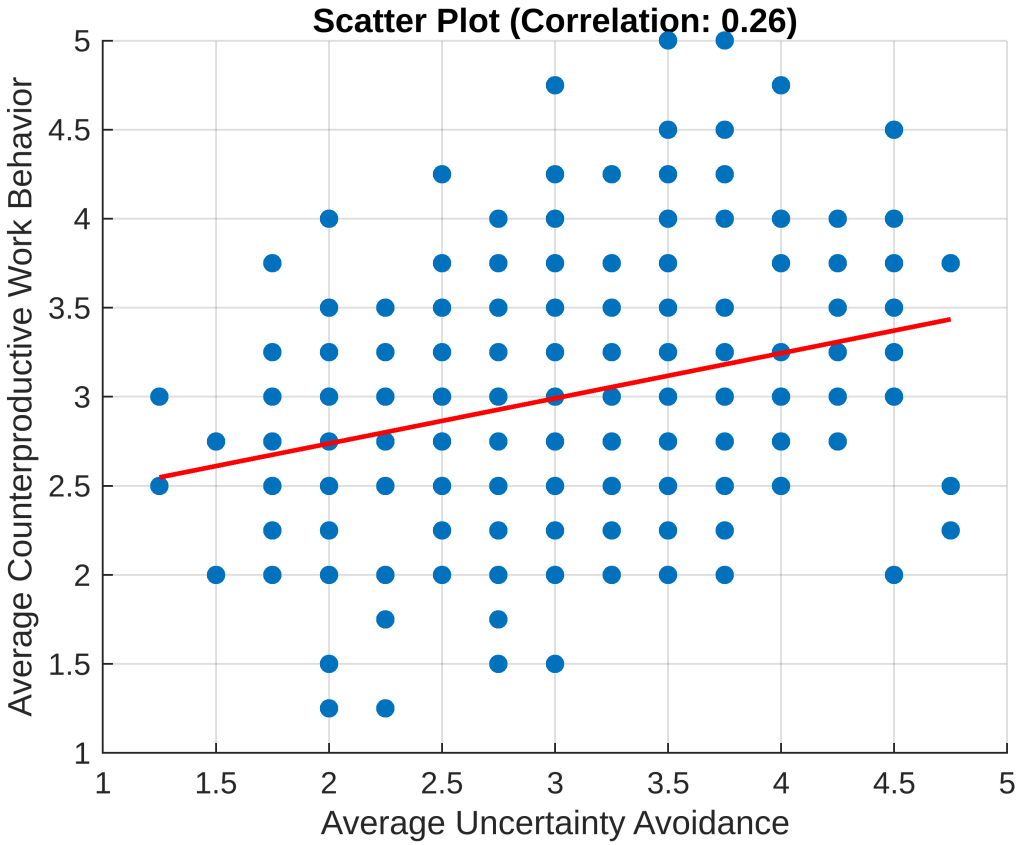



```
% Compute the correlation
correlationValue = corr(averageUncertaintyAvoidance,
averageCounterproductiveWorkBehavior)
```

```
correlationValue =
0.2608
```

```
% Scatter plot with trend line
scatter(averageUncertaintyAvoidance, averageCounterproductiveWorkBehavior,
'filled');
hold on;
p = polyfit(averageUncertaintyAvoidance,
averageCounterproductiveWorkBehavior, 1); % Fit line
x_fit = linspace(min(averageUncertaintyAvoidance),
max(averageUncertaintyAvoidance), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, '-r', 'LineWidth', 1.5); % Trend line
hold off;

xlabel('Average Uncertainty Avoidance');
ylabel('Average Counterproductive Work Behavior');
title(['Scatter Plot (Correlation: ', num2str(correlationValue, '%.2f'),
')']);
grid on;
```




```
% Combine all variables into a matrix
dataMatrix = [averagePower_Distance, averageIndividualismvsCollectivism,
averageMasculinityvsFemininity, averageUncertaintyAvoidance,
averageLong_termvsShort_termOrientation, averageTaskPerformance,
averageContextualPerformance, averageAdaptivePerformance,
averageCounterproductiveWorkBehavior];

% Compute the correlation matrix
correlationMatrix = corr(dataMatrix);

% Display the correlation matrix
disp('Correlation Matrix:');
```

```
Correlation Matrix:
```

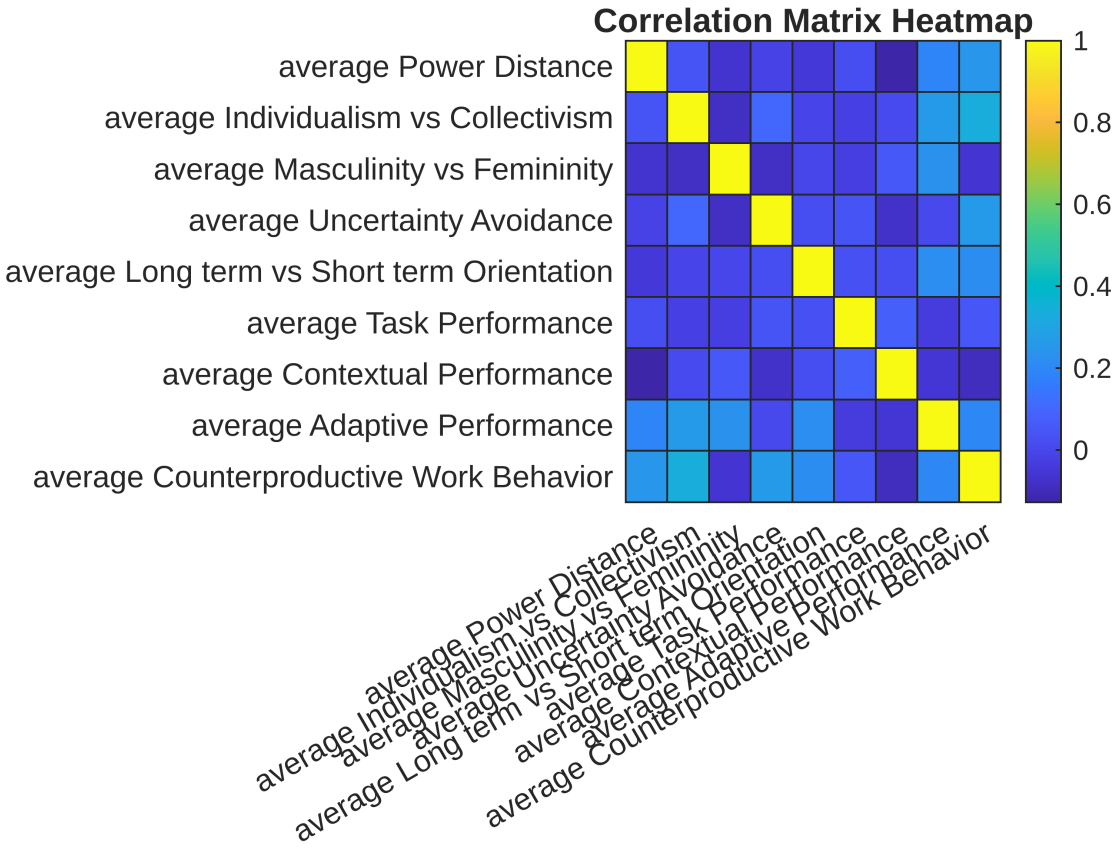
```
disp(correlationMatrix);
```

```
    1.0000    0.0413   -0.0625   -0.0169   -0.0473    0.0237   -0.1272    0.1942    0.2509
    0.0413    1.0000   -0.0827    0.0977   -0.0055   -0.0233    0.0111    0.2611    0.3391
   -0.0625   -0.0827    1.0000   -0.0798   -0.0002   -0.0267    0.0592    0.2339   -0.0609
   -0.0169    0.0977   -0.0798    1.0000    0.0261    0.0407   -0.0717    0.0055    0.2608
   -0.0473   -0.0055   -0.0002    0.0261    1.0000    0.0274    0.0246    0.2245    0.2163
    0.0237   -0.0233   -0.0267    0.0407    0.0274    1.0000    0.0763   -0.0371    0.0507
   -0.1272    0.0111    0.0592   -0.0717    0.0246    0.0763    1.0000   -0.0531   -0.0877
    0.1942    0.2611    0.2339    0.0055    0.2245   -0.0371   -0.0531    1.0000    0.2026
    0.2509    0.3391   -0.0609    0.2608    0.2163    0.0507   -0.0877    0.2026    1.0000
```

```
% Variable labels
```

```
variableNames = {'average Power Distance', 'average Individualism vs
Collectivism', 'average Masculinity vs Femininity', 'average Uncertainty
Avoidance', 'average Long term vs Short term Orientation', 'average
Task Performance', 'average Contextual Performance', 'average Adaptive
Performance', 'average Counterproductive Work Behavior'};

% Create a heatmap
figure;
heatmap(variableNames, variableNames, correlationMatrix, 'Colormap', parula,
'ColorbarVisible', 'on');
title('Correlation Matrix Heatmap');
```




```
% Perform linear regression
regressionModel = fitlm(averageAutocraticLeadership, averageTaskPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)      2.7863     0.25388     10.975    9.2494e-24
    x1             0.057068    0.083513    0.68334       0.49492


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.708
```

```
R-squared: 0.00156,  Adjusted R-Squared: -0.00179
F-statistic vs. constant model: 0.467, p-value = 0.495
```

```
% Perform linear regression
regressionModel = fitlm(averageAutocraticLeadership,
averageContextualPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate        SE        tStat        pValue
                   _________    _________    _________    __________

    (Intercept)       3.2168      0.25961       12.391    1.0098e-28
    x1             -0.043656     0.085395     -0.51122       0.60957


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.724
R-squared: 0.000876,  Adjusted R-Squared: -0.00248
F-statistic vs. constant model: 0.261, p-value = 0.61
```

```
% Perform linear regression
regressionModel = fitlm(averageAutocraticLeadership,
averageAdaptivePerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   _________    ________    _______    __________

    (Intercept)      2.7751      0.26529     10.461    5.0964e-22
    x1             0.085219     0.087266    0.97655       0.32959


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.74
R-squared: 0.00319,  Adjusted R-Squared: -0.000155
F-statistic vs. constant model: 0.954, p-value = 0.33
```

```
% Perform linear regression
regressionModel = fitlm(averageAutocraticLeadership,
averageCounterproductiveWorkBehavior)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate        SE        tStat       pValue
                   _________    _________    _______    __________

    (Intercept)       3.0651      0.25372     12.081    1.2868e-27
    x1             -0.020022     0.083458    -0.2399       0.81057
```

```
Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.707
R-squared: 0.000193,  Adjusted R-Squared: -0.00316
F-statistic vs. constant model: 0.0576, p-value = 0.811
```

```
% Perform linear regression
regressionModel = fitlm(averageAutocraticLeadership, averagePower_Distance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat        pValue
                   ________    _______    ________    __________

    (Intercept)       3.0583    0.24433      12.517    3.5719e-29
    x1             0.0024958    0.08037    0.031054       0.97525


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.681
R-squared: 3.24e-06,  Adjusted R-Squared: -0.00335
F-statistic vs. constant model: 0.000964, p-value = 0.975
```

```
% Perform linear regression
regressionModel = fitlm(averageAutocraticLeadership,
averageIndividualismvsCollectivism)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)      3.3342     0.24897     13.392    2.4391e-32
    x1              -0.11917    0.081898    -1.4551        0.1467


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.694
R-squared: 0.00705,  Adjusted R-Squared: 0.00372
F-statistic vs. constant model: 2.12, p-value = 0.147
```

```
% Perform linear regression
regressionModel = fitlm(averageAutocraticLeadership,
averageMasculinityvsFemininity)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE         tStat       pValue
                   _________    ________    _________    ________

    (Intercept)       2.9857      0.2698       11.066    4.48e-24
    x1             -0.0030049    0.088747    -0.033859     0.97301
```

```
Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.752
R-squared: 3.85e-06,  Adjusted R-Squared: -0.00335
F-statistic vs. constant model: 0.00115, p-value = 0.973
```

```
% Perform linear regression
regressionModel = fitlm(averageAutocraticLeadership,
averageUncertaintyAvoidance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)       3.752     0.25765     14.562    1.1918e-36
    x1              -0.23285    0.084753    -2.7474     0.0063722


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.718
R-squared: 0.0247,  Adjusted R-Squared: 0.0214
F-statistic vs. constant model: 7.55, p-value = 0.00637
```

```
% Perform linear regression
regressionModel = fitlm(averageAutocraticLeadership,
averageLong_termvsShort_termOrientation)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat        pValue
                   ________    ________    ________    __________

    (Intercept)      3.1265     0.26276      11.899    5.6563e-27
    x1              -0.03744    0.086431    -0.43318        0.6652


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.733
R-squared: 0.000629,  Adjusted R-Squared: -0.00272
F-statistic vs. constant model: 0.188, p-value = 0.665
```

```
% Perform linear regression
regressionModel = fitlm(averagePower_Distance, averageTaskPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)      2.8821     0.18911      15.24    3.5533e-39
    x1             0.024605    0.060224    0.40855       0.68316
```

```
Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.708
R-squared: 0.00056,  Adjusted R-Squared: -0.00279
F-statistic vs. constant model: 0.167, p-value = 0.683
```

```
% Perform linear regression
regressionModel = fitlm(averagePower_Distance, averageContextualPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)      3.5004     0.19179     18.251    1.7544e-50
    x1              -0.13522    0.061077    -2.2139      0.027595


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.718
R-squared: 0.0162,  Adjusted R-Squared: 0.0129
F-statistic vs. constant model: 4.9, p-value = 0.0276
```

```
% Perform linear regression
regressionModel = fitlm(averagePower_Distance, averageAdaptivePerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate      SE       tStat       pValue
                   ________    _______    ______    __________

    (Intercept)     2.3833     0.19406    12.281    2.4909e-28
    x1              0.21122     0.0618    3.4178    0.00071917


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.727
R-squared: 0.0377,  Adjusted R-Squared: 0.0345
F-statistic vs. constant model: 11.7, p-value = 0.000719
```

```
% Perform linear regression
regressionModel = fitlm(averagePower_Distance,
averageCounterproductiveWorkBehavior)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    ______    __________

    (Intercept)     2.2062      0.18286    12.064    1.4696e-27
    x1              0.26056     0.058235    4.4744    1.0918e-05
```

```
Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.685
R-squared: 0.063,  Adjusted R-Squared: 0.0598
F-statistic vs. constant model: 20, p-value = 1.09e-05
```

```
% Perform linear regression
regressionModel = fitlm(averageIndividualismvsCollectivism,
averageTaskPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate        SE        tStat        pValue
                   ________    ________    ________    __________

    (Intercept)       3.028     0.18001      16.821    4.1711e-45
    x1             -0.023671    0.058893    -0.40193       0.68802


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.708
R-squared: 0.000542,  Adjusted R-Squared: -0.00281
F-statistic vs. constant model: 0.162, p-value = 0.688
```

```
% Perform linear regression
regressionModel = fitlm(averageIndividualismvsCollectivism,
averageContextualPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)      3.0516     0.18404     16.581    3.3334e-44
    x1             0.011504    0.060212    0.19106       0.84861


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.724
R-squared: 0.000122,  Adjusted R-Squared: -0.00323
F-statistic vs. constant model: 0.0365, p-value = 0.849
```

```
% Perform linear regression
regressionModel = fitlm(averageIndividualismvsCollectivism,
averageAdaptivePerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    ______    __________

    (Intercept)     2.2042      0.18177    12.126    8.8656e-28
```

```
    x1              0.27771     0.059468    4.6699    4.5647e-06


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.715
R-squared: 0.0682,  Adjusted R-Squared: 0.0651
F-statistic vs. constant model: 21.8, p-value = 4.56e-06
```

```
% Perform linear regression
regressionModel = fitlm(averageIndividualismvsCollectivism,
averageCounterproductiveWorkBehavior)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE       tStat       pValue
                   ________    ________    ______    __________

    (Intercept)     1.9801      0.16916    11.705    2.7195e-26
    x1              0.34433     0.055344    6.2215    1.6645e-09


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.665
R-squared: 0.115,  Adjusted R-Squared: 0.112
F-statistic vs. constant model: 38.7, p-value = 1.66e-09
```

```
% Perform linear regression
regressionModel = fitlm(averageMasculinityvsFemininity,
averageTaskPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate        SE        tStat        pValue
                   _________    ________    ________    __________

    (Intercept)       3.0323      0.1674      18.114    5.7652e-50
    x1             -0.025121    0.054535    -0.46063        0.6454


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.708
R-squared: 0.000712,  Adjusted R-Squared: -0.00264
F-statistic vs. constant model: 0.212, p-value = 0.645
```

```
% Perform linear regression
regressionModel = fitlm(averageMasculinityvsFemininity,
averageContextualPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE       tStat       pValue
                   ________    ________    ______    __________
```

```
    (Intercept)      2.9163     0.17088    17.067    4.9716e-46
    x1             0.056949    0.055667     1.023       0.30712


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.723
R-squared: 0.0035,  Adjusted R-Squared: 0.000156
F-statistic vs. constant model: 1.05, p-value = 0.307
```

```
% Perform linear regression
regressionModel = fitlm(averageMasculinityvsFemininity,
averageAdaptivePerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate      SE       tStat       pValue
                   ________    _______    ______    __________

    (Intercept)      2.345     0.17027    13.772    9.8474e-34
    x1             0.23039     0.05547    4.1535    4.2814e-05


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.72
R-squared: 0.0547,  Adjusted R-Squared: 0.0516
F-statistic vs. constant model: 17.3, p-value = 4.28e-05
```

```
% Perform linear regression
regressionModel = fitlm(averageMasculinityvsFemininity,
averageCounterproductiveWorkBehavior)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   _________    _______    _______    __________

    (Intercept)       3.1754    0.16692     19.023    2.2483e-53
    x1             -0.057255    0.05438    -1.0529       0.29325


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.706
R-squared: 0.00371,  Adjusted R-Squared: 0.000363
F-statistic vs. constant model: 1.11, p-value = 0.293
```

```
% Perform linear regression
regressionModel = fitlm(averageUncertaintyAvoidance, averageTaskPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
```

(Intercept)	2.8366	0.1769	16.035	3.7456e-42
x1	0.039598	0.056369	0.70248	0.48293

Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.708
R-squared: 0.00165, Adjusted R-Squared: -0.0017
F-statistic vs. constant model: 0.493, p-value = 0.483

```
% Perform linear regression
regressionModel = fitlm(averageUncertaintyAvoidance,
averageContextualPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
```

	Estimate	**SE**	**tStat**	**pValue**
(Intercept)	3.3037	0.18051	18.302	1.1287e-50
x1	-0.071352	0.057519	-1.2405	0.21577

Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.722
R-squared: 0.00514, Adjusted R-Squared: 0.0018
F-statistic vs. constant model: 1.54, p-value = 0.216

```
% Perform linear regression
regressionModel = fitlm(averageUncertaintyAvoidance,
averageAdaptivePerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
```

	Estimate	**SE**	**tStat**	**pValue**
(Intercept)	3.0138	0.18515	16.278	4.6106e-43
x1	0.0055927	0.058998	0.094794	0.92454

Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.741
R-squared: 3.02e-05, Adjusted R-Squared: -0.00333
F-statistic vs. constant model: 0.00899, p-value = 0.925

```
% Perform linear regression
regressionModel = fitlm(averageUncertaintyAvoidance,
averageCounterproductiveWorkBehavior)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1
```

```
Estimated Coefficients:
                   Estimate      SE       tStat       pValue
                   ________    ________    ______    __________

    (Intercept)     2.2306     0.17069    13.068    3.6709e-31
    x1             0.25362     0.05439     4.663    4.7099e-06


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.683
R-squared: 0.068,  Adjusted R-Squared: 0.0649
F-statistic vs. constant model: 21.7, p-value = 4.71e-06
```

```
% Perform linear regression
regressionModel = fitlm(averageLong_termvsShort_termOrientation,
averageTaskPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate      SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)      2.8776     0.17361     16.575    3.5143e-44
    x1             0.026519    0.055978    0.47374       0.63603


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.708
R-squared: 0.000753,  Adjusted R-Squared: -0.0026
F-statistic vs. constant model: 0.224, p-value = 0.636
```

```
% Perform linear regression
regressionModel = fitlm(averageLong_termvsShort_termOrientation,
averageContextualPerformance)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate      SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)      3.0126     0.17747     16.975    1.0989e-45
    x1             0.024282    0.057224    0.42434       0.67163


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.724
R-squared: 0.000604,  Adjusted R-Squared: -0.00275
F-statistic vs. constant model: 0.18, p-value = 0.672
```

```
% Perform linear regression
regressionModel = fitlm(averageLong_termvsShort_termOrientation,
averageAdaptivePerformance)
```

```
regressionModel =
```

```
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE       tStat       pValue
                   ________    ________    ______    __________

    (Intercept)     2.3469      0.17699     13.26    7.3918e-32
    x1             0.22691     0.057069    3.9761    8.7976e-05


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.722
R-squared: 0.0504,  Adjusted R-Squared: 0.0472
F-statistic vs. constant model: 15.8, p-value = 8.8e-05
```

```
% Perform linear regression
regressionModel = fitlm(averageLong_termvsShort_termOrientation,
averageCounterproductiveWorkBehavior)
```

```
regressionModel =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE       tStat       pValue
                   ________    ________    ______    __________

    (Intercept)     2.3755      0.16933    14.029    1.1231e-34
    x1             0.20885     0.054599    3.8251    0.00015922


Number of observations: 300, Error degrees of freedom: 298
Root Mean Squared Error: 0.691
R-squared: 0.0468,  Adjusted R-Squared: 0.0436
F-statistic vs. constant model: 14.6, p-value = 0.000159
```