import pandas as pd

import matplotlib.pyplot as plt

import seaborn

import numpy as np

df = pd.read\_csv('studentsperformance.csv')

print(df)

gender race/ethnicity parental level of education lunch \

0 female group B bachelor's degree standard

1 female group C some college standard

2 female group B master's degree standard

3 male group A associate's degree free/reduced

4 male group C some college standard

.. ... ... ... ...

995 female group E master's degree standard

996 male group C high school free/reduced

997 female group C high school free/reduced

998 female group D some college standard

999 female group D some college free/reduced

test preparation course math score reading score writing score

0 none 72 72 74

1 completed 69 90 88

2 none 90 95 93

3 none 47 57 44

4 none 76 78 75

.. ... ... ... ...

995 completed 88 99 95

996 none 62 55 55

997 completed 59 71 65

998 completed 68 78 77

999 none 77 86 86

[1000 rows x 8 columns]

# Generate some data for the Histogram plots

gender = df['gender']

race\_ethnicity = df['race/ethnicity']

parental\_level\_of\_education = df['parental level of education']

lunch = df['lunch']

test\_preparation\_course = df['test preparation course']

math\_score = df['math score']

reading\_score = df['reading score']

writing\_score = df['writing score']

#Histogram

# Create the histogram plot using Seaborn

seaborn.histplot(data=df['reading score'],stat='count',color='black',kde=True)

# Customize the appearance of the plot

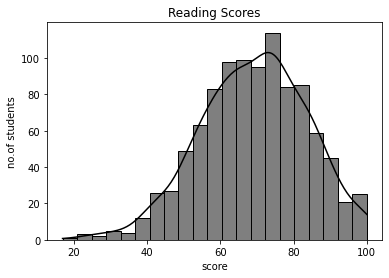
plt.xlabel("score")

plt.ylabel("no.of students")

plt.title("Reading Scores")

# Show the plot

plt.show()



# Create the histogram plot

plt.hist(df['math score'],histtype="stepfilled")

# Customize the appearance of the plot

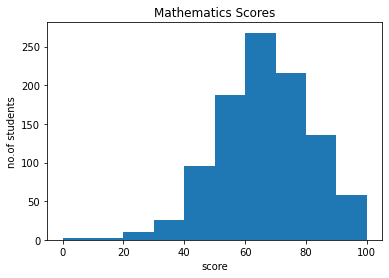
plt.xlabel("score")

plt.ylabel("no.of students")

plt.title("Mathematics Scores")

# Show the plot

plt.show()



# Create the histogram plot

plt.hist(writing\_score, bins=25, alpha=0.5, label='writing ', color='red')

plt.hist(reading\_score, bins=25, alpha=0.5, label='Reading', color='yellow')

plt.hist(math\_score, bins=25, alpha=0.5, label='Mathematics', color='blue')

# Customize the plot

plt.xlabel('Scores')

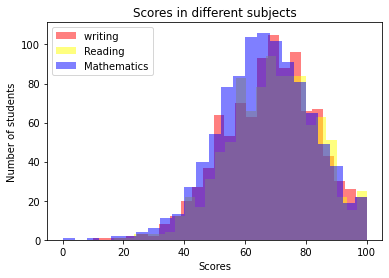
plt.ylabel('Number of students')

plt.title('Scores in different subjects')

plt.legend()

# Show the plot

plt.show()



# Create a figure and axis with a larger size

plt.figure(figsize=(30, 15))

# Create the histogram plot for catagorical data columns

plt.hist(lunch, bins=2, alpha=0.5, label='lunch', color='red')

plt.hist(gender, bins=2, alpha=0.5, label = 'gender', color='blue')

plt.hist(test\_preparation\_course, bins=2, alpha=0.5, label = 'test preparation course', color='green')

plt.hist(parental\_level\_of\_education, bins=6, alpha=0.5, label = 'parental level of education', color='orange')

plt.hist(race\_ethnicity, bins=5, alpha=0.5, label = 'race/ethnicity', color='black')

# Customize the plot

plt.title("Catagorical Data columns", size="25")

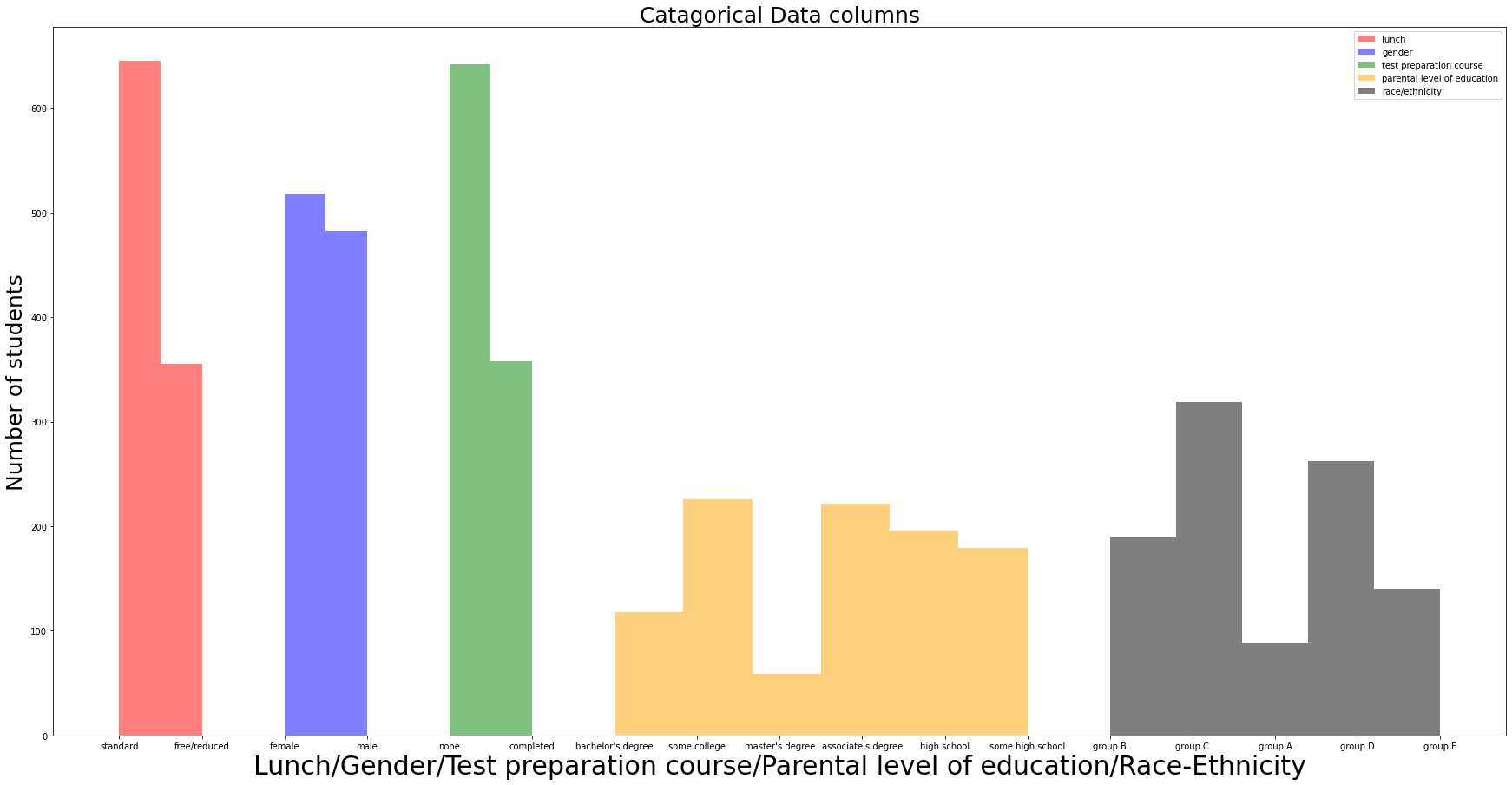
plt.ylabel("Number of students",size='25')

plt.xlabel("Lunch/Gender/Test preparation course/Parental level of education/Race-Ethnicity",size="30")

plt.legend()

#show the plot

plt.show()



#BOXPLOT

# Calculate the values you want to label

quartiles = list(np.percentile(writing\_score, [25, 50, 75]))

min\_value = min(writing\_score)

max\_value = max(writing\_score)

IQR = quartiles[2] - quartiles[0]

ul = quartiles[2]+1.5\*IQR

ll = quartiles[0]-1.5\*IQR

if(max\_value < ul):

ul = max\_value

if(ll < min\_value):

ll = min\_value

# Create a clear and big box plot

plt.figure(figsize=(10, 6)) # Set the figure size

# Create the box plot

plt.boxplot(math\_score)

# Label the statistics on the box plot

plt.text(1.1, quartiles[0], f'Q1: {quartiles[0]}', va='top', color='red')

plt.text(1.1, quartiles[1], f'Q2: {quartiles[1]}', va='center', color='red')

plt.text(1.1, quartiles[2], f'Q3: {quartiles[2]}', va='center', color='red')

plt.text(1.05, ul,f'Upperlimit: {ul}',va='center', color='red')

plt.text(1.05, ll,f'Lowerlimit: {ll}',va='center', color='red')

plt.text(1.01, min\_value, f'Min: {min\_value}', va='center', color='red')

plt.text(1.2, max\_value, f',Max: {max\_value}', va='center', color='red')

# Customize the appearance of the plot

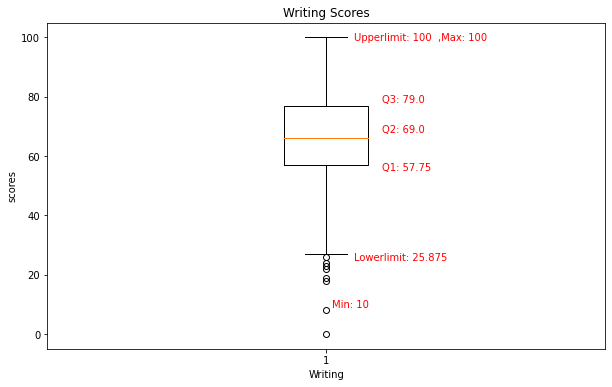
plt.xlabel("Writing")

plt.ylabel("scores")

plt.title("Writing Scores")

# Show the box plot

plt.show()



# Calculate the values you want to label

quartiles = list(np.percentile(reading\_score, [25, 50, 75]))

min\_value = min(reading\_score)

max\_value = max(reading\_score)

IQR = quartiles[2] - quartiles[0]

ul = quartiles[2]+1.5\*IQR

ll = quartiles[0]-1.5\*IQR

if(max\_value < ul):

ul = max\_value

if(ll < min\_value):

ll = min\_value

# Create a clear and big box plot using Seaborn

plt.figure(figsize=(10, 6)) # Set the figure size

# Create the box plot

seaborn.boxplot(data=reading\_score, width=0.4)

# Label the statistics on the box plot

plt.text(0.21, quartiles[0], f'Q1: {quartiles[0]}', va='top', color='red')

plt.text(0.21, quartiles[1], f'Q2: {quartiles[1]}', va='center', color='red')

plt.text(0.21, quartiles[2], f'Q3: {quartiles[2]}', va='center', color='red')

plt.text(0.15, ul,f'Upperlimit: {ul}',va='center', color='red')

plt.text(0.15, ll,f'Lowerlimit: {ll}',va='center', color='red')

plt.text(0.1, min\_value, f'Min: {min\_value}', va='center', color='red')

plt.text(0.3, max\_value, f',Max: {max\_value}', va='center', color='red')

# Customize the appearance of the plot

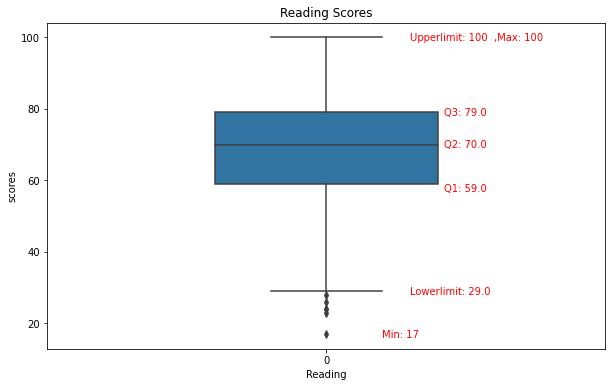
plt.xlabel("Reading")

plt.ylabel("scores")

plt.title("Reading Scores")

# Show the box plot

plt.show()



# Calculate the values you want to label

quartiles = list(np.percentile(math\_score, [25, 50, 75]))

min\_value = min(math\_score)

max\_value = max(math\_score)

IQR = quartiles[2] - quartiles[0]

ul = quartiles[2]+1.5\*IQR

ll = quartiles[0]-1.5\*IQR

if(max\_value < ul):

ul = max\_value

if(ll < min\_value):

ll = min\_value

# Create a clear and big box plot using Seaborn

plt.figure(figsize=(10, 6)) # Set the figure size

# Create the box plot

seaborn.boxplot(data=math\_score, width=0.4)

# Label the statistics on the box plot

plt.text(0.21, quartiles[0], f'Q1: {quartiles[0]}', va='top', color='red')

plt.text(0.21, quartiles[1], f'Q2: {quartiles[1]}', va='center', color='red')

plt.text(0.21, quartiles[2], f'Q3: {quartiles[2]}', va='center', color='red')

plt.text(0.15, ul,f'Upperlimit: {ul}',va='center', color='red')

plt.text(0.15, ll,f'Lowerlimit: {ll}',va='center', color='red')

plt.text(0.21, min\_value, f'Min: {min\_value}', va='center', color='red')

plt.text(0.3, max\_value, f',Max: {max\_value}', va='center', color='red')

# Customize the appearance of the plot

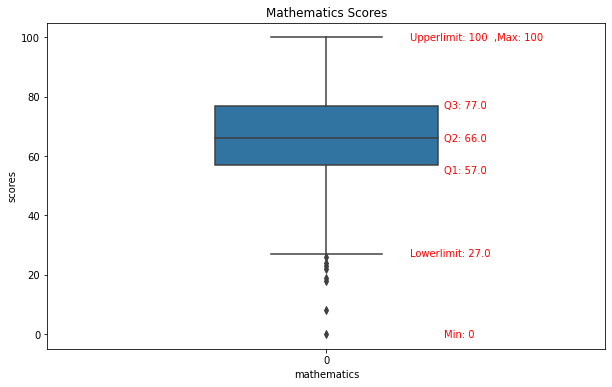
plt.xlabel("mathematics")

plt.ylabel("scores")

plt.title("Mathematics Scores")

# Show the box plot

plt.show()



# Create a figure with a larger size

plt.figure(figsize=(25, 10))

data = pd.DataFrame({'math': math\_score, 'Reading': reading\_score, 'writing': writing\_score})

# Create a boxplot for multiple data sets using Seaborn

seaborn.boxplot(data=data, width=0.4)

# Customize the appearance of the plot

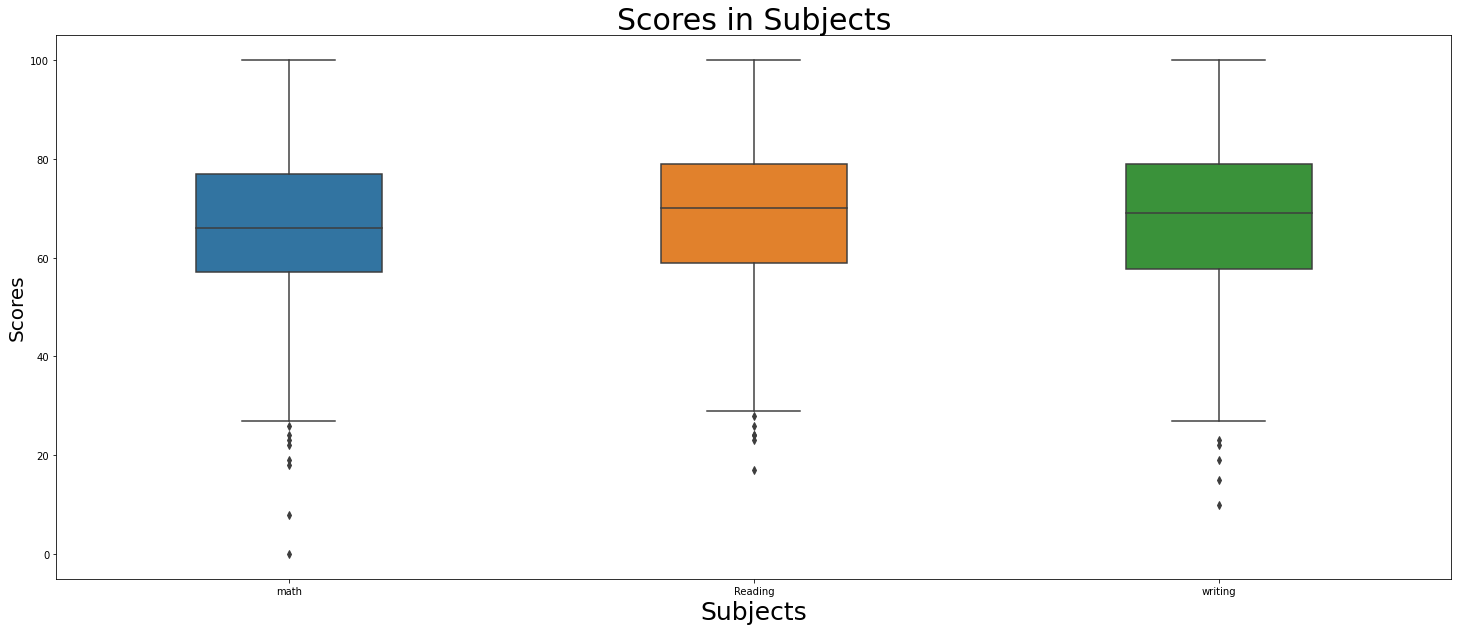
plt.xlabel('Subjects',size="25")

plt.ylabel('Scores', size='20')

plt.title('Scores in Subjects', size="30")

# Show the plot

plt.show()



#LINEPLOT

# Generate some data for the line plot

x = np.linspace(0, 10, 10) # Create an array of 10 points from 0 to 10

# Create a clear and big box plot using Seaborn

plt.figure(figsize=(15, 6))# Set the figure size

# Create the line plot

plt.plot(x, math\_score.head(10), label='math scores for starting rows', color='blue', linestyle='-', linewidth=2)

plt.plot(x, math\_score.tail(10), label='math scores for last rows', color='red', linestyle='-', linewidth=2)

# Customize the appearance of the plot

plt.title('Math score Line Plot')

plt.xlabel('Rows', size="15")

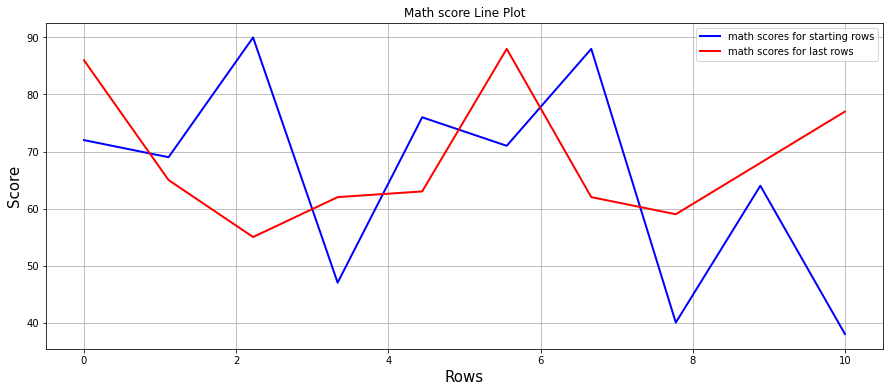
plt.ylabel('Score', size="15")

plt.grid(True)

plt.legend()

# Show the plot

plt.show()



# Generate some data for the line plot

x = np.linspace(0, 10, 10) # Create an array of 10 points from 0 to 10

# Create a clear and big box plot using Seaborn

plt.figure(figsize=(15, 6))# Set the figure size

# Create the line plot

plt.plot(x, writing\_score.head(10), label='Writing scores for starting rows', color='yellow', linestyle='--', linewidth=5)

plt.plot(x, writing\_score.tail(10), label='Writing scores for last rows', color='orange', linestyle='--', linewidth=5)

# Customize the appearance of the plot

plt.title('Writing scores Line Plot')

plt.xlabel('Rows', size="15")

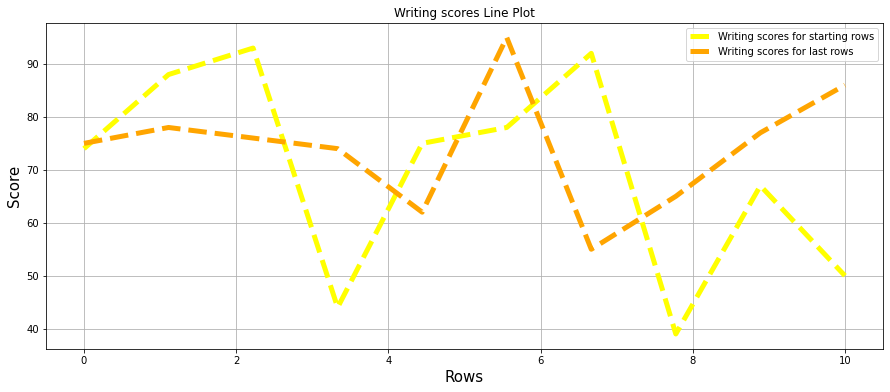
plt.ylabel('Score', size="15")

plt.grid(True)

plt.legend()

# Show the plot

plt.show()



# Generate some data for the line plot

x = np.linspace(0, 10, 10) # Create an array of 10 points from 0 to 10

# Create a clear and big box plot using Seaborn

plt.figure(figsize=(15, 6))# Set the figure size

# Create the line plot

plt.plot(x, reading\_score.head(10), label='Reading scores for starting rows', color='green', linestyle=':', linewidth=5)

plt.plot(x, reading\_score.tail(10), label='Reading scores for last rows', color='black', linestyle=':', linewidth=5)

# Customize the appearance of the plot

plt.title('Reading scores Line Plot')

plt.xlabel('Rows', size="15")

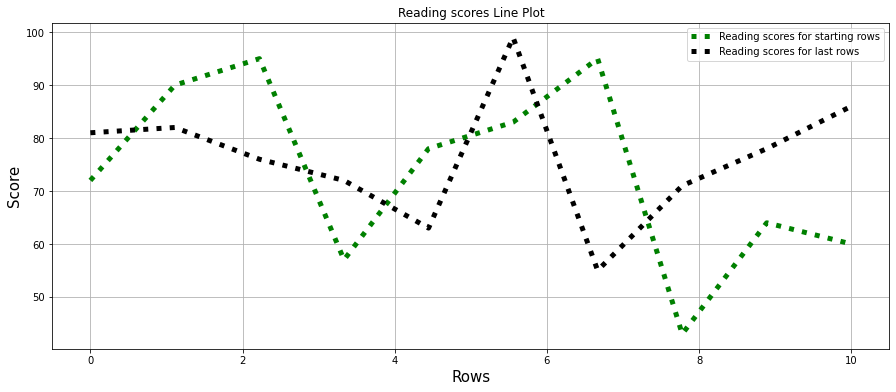
plt.ylabel('Score', size="15")

plt.grid(True)

plt.legend()

# Show the plot

plt.show()



*#plt.plot(x, writing\_score.tail(10), label='Reading scores for last rows', color='black', linestyle=':', linewidth=5)*

*#Supported values are '-', '--', '-.', ':', 'None', ' ', '', 'solid', 'dashed', 'dashdot', 'dotted'*

*# Generate some data for the line plot*

x = np.linspace(0, 10, 10) # Create an array of 10 points from 0 to 10

*# Create a clear and big box plot using Seaborn*

plt.figure(figsize=(15, 6))# Set the figure size

*# Create the line plot*

plt.plot(x, writing\_score.head(10), label='Reading scores', color='green', linestyle='-', linewidth=2)

# Create the line plot using Seaborn

seaborn.lineplot(x=x,y= reading\_score.head(10), label='Reading scores', color='blue', linestyle='--', linewidth=2)

seaborn.lineplot(x=x,y= math\_score.head(10), label='Math scores', color='red', linestyle='-.', linewidth=2)

# Customize the appearance of the plot

plt.title('Starting rows scores Line Plot')

plt.xlabel('Rows', size="15")

plt.ylabel('Score', size="15")

plt.legend()

# Show the plot

plt.show()