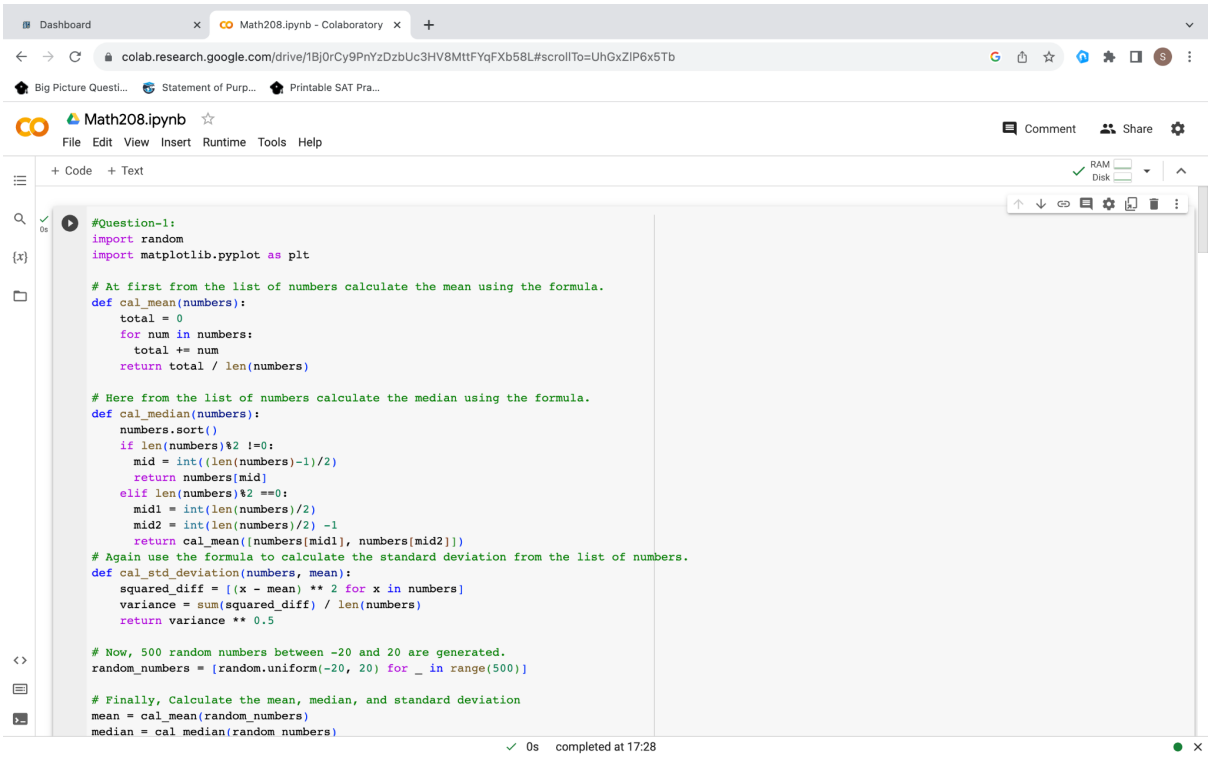
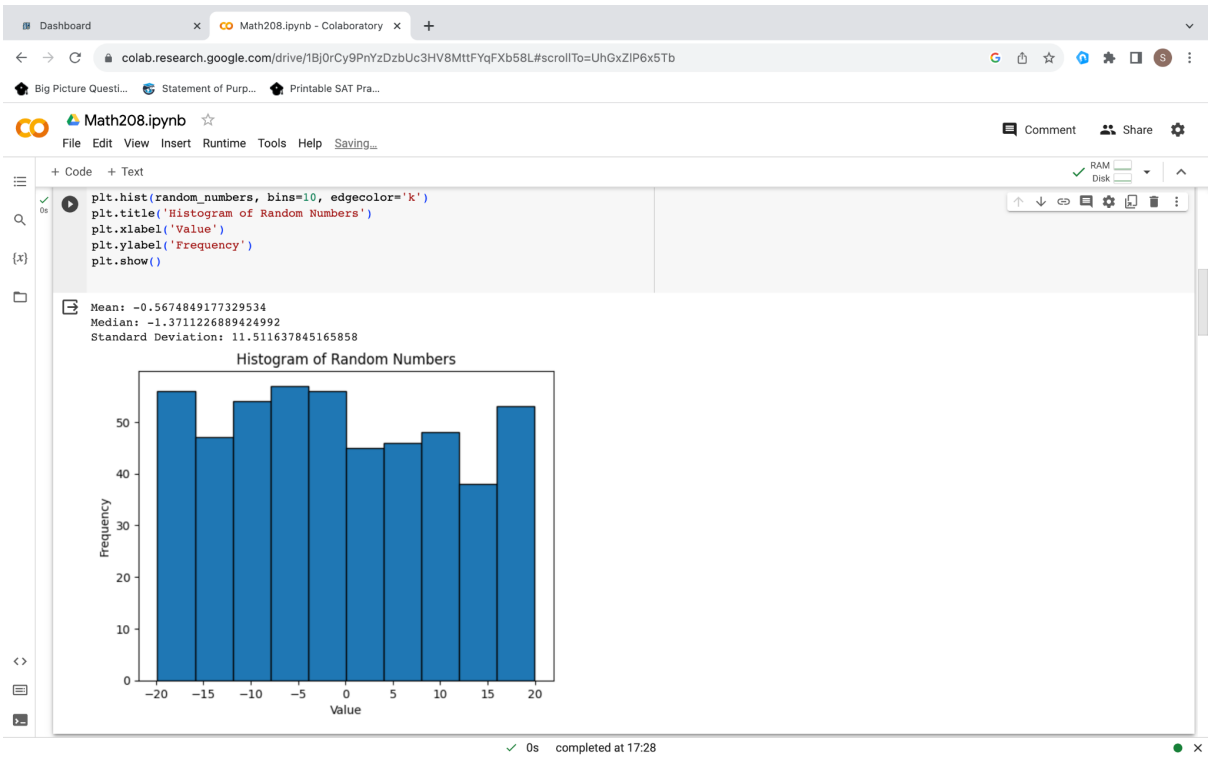
Week-2 Assignment 1
Name- Swagat Neupane
ID- 19698ns
San Francisco Bay university
Course : MATH208 - Probability and Statistics





Chrome File Edit View History Bookmarks Profiles Tab Window Help Sun Oct 1 5:28 PM

Dashboard | Math208.ipynb - Colaboratory

colab.research.google.com/drive/1Bj0rCy9PnYzDzbUc3HV8MttFYqFXb58L#scrollTo=UhGxZlP6x5Tb

Big Picture Questi... Statement of Purp... Printable SAT Pra...

Math208.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

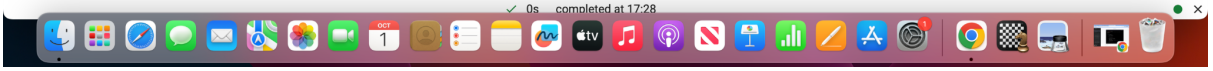
```
#Question-2
import random
import matplotlib.pyplot as plt
import math

# At first from the list of numbers calculate the mean using the formula.
def cal_mean(numbers):
    total = 0
    for num in numbers:
      total += num
    return total / len(numbers)

# Here from the list of numbers calculate the median using the formula.
def cal_median(numbers):
    numbers.sort()
    if len(numbers)%2 !=0:
      mid = int((len(numbers)-1)/2)
      return numbers[mid]
    elif len(numbers)%2 ==0:
      mid1 = int(len(numbers)/2)
      mid2 = int(len(numbers)/2) -1
      return cal_mean([numbers[mid1], numbers[mid2]])
# Again use the formula to calculate the standard deviation from the list of numbers.
def cal_std_deviation(numbers, mean):
    squared_diff = [(x - mean) ** 2 for x in numbers]
    variance = sum(squared_diff) / len(numbers)
    return variance ** 0.5

# Generate 500 random numbers with mean = 10 and standard deviation = 0.5
```

0s completed at 17:28



Chrome File Edit View History Bookmarks Profiles Tab Window Help Sun Oct 1 5:29 PM

Dashboard | Math208.ipynb - Colaboratory

colab.research.google.com/drive/1Bj0rCy9PnYzDzbUc3HV8MttFYqFXb58L#scrollTo=tLuEZTu4yC4v

Big Picture Questi... Statement of Purp... Printable SAT Pra...

Math208.ipynb

File Edit View Insert Runtime Tools Help All changes saved

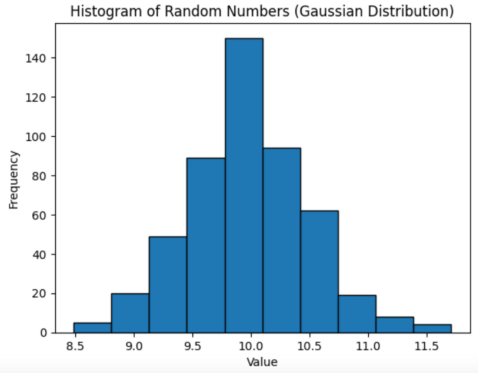
Comment Share

+ Code + Text

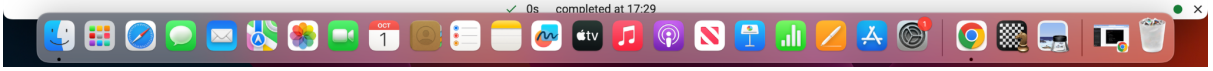
RAM Disk

```
plt.show()
```

```
Mean: 9.97721916136183
Median: 9.984634380438285
Standard Deviation: 0.5090161099623043
```



0s completed at 17:29



Chrome File Edit View History Bookmarks Profiles Tab Window Help Sun Oct 1 5:29 PM

Dashboard | Math208.ipynb - Colaboratory

colab.research.google.com/drive/1Bj0rCy9PnYzDzbUc3HV8MttFYqFXb58L#scrollTo=Lh0P_LhjyFvi

Big Picture Questi... Statement of Purp... Printable SAT Pra...

Math208.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk

```
#Question-3:
import matplotlib.pyplot as plt

# Data for the leading causes of death
causes = [
    "Heart Disease", "Malignant Neoplasms", "Stroke",
    "Chronic Respiratory Disease", "Accidents",
    "Diabetes", "Alzheimer's", "Influenza/Pneumonia",
    "Nephritis/Nephrosis", "Septicemia"
]

deaths = [
    63.2, 56.0, 13.7, 12.5, 12.2,
    7.2, 7.2, 5.6, 4.5, 3.4
]

# Sort causes and deaths in descending order
sorted_causes, sorted_deaths = zip(*sorted(zip(causes, deaths), key=lambda x: x[1], reverse=True))

# Calculate the cumulative percentage
total_deaths = sum(sorted_deaths)
cumulative_percentage = [100 * sum(sorted_deaths[:i+1]) / total_deaths for i in range(len(sorted_deaths))]

# Create the Pareto chart
fig, ax1 = plt.subplots()

ax1.bar(sorted_causes, sorted_deaths, color='b', alpha=0.7)
ax1.set_xlabel('Cause of Death')
ax1.set_ylabel('Number of Deaths (x 10,000)', color='b')
```

0s completed at 17:29

Chrome File Edit View History Bookmarks Profiles Tab Window Help Sun Oct 1 5:29 PM

Dashboard | Math208.ipynb - Colaboratory

colab.research.google.com/drive/1Bj0rCy9PnYzDzbUc3HV8MttFYqFXb58L#scrollTo=Lh0P_LhjyFvi

Big Picture Questi... Statement of Purp... Printable SAT Pra...

Math208.ipynb

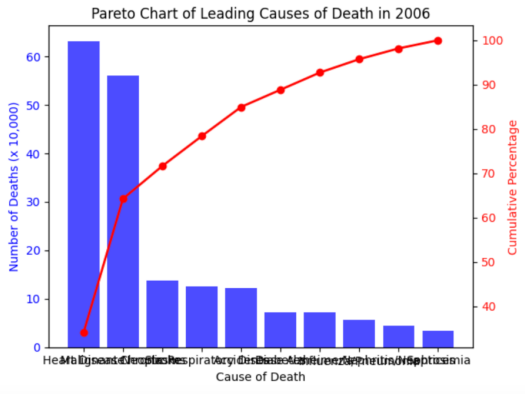
File Edit View Insert Runtime Tools Help

Comment Share

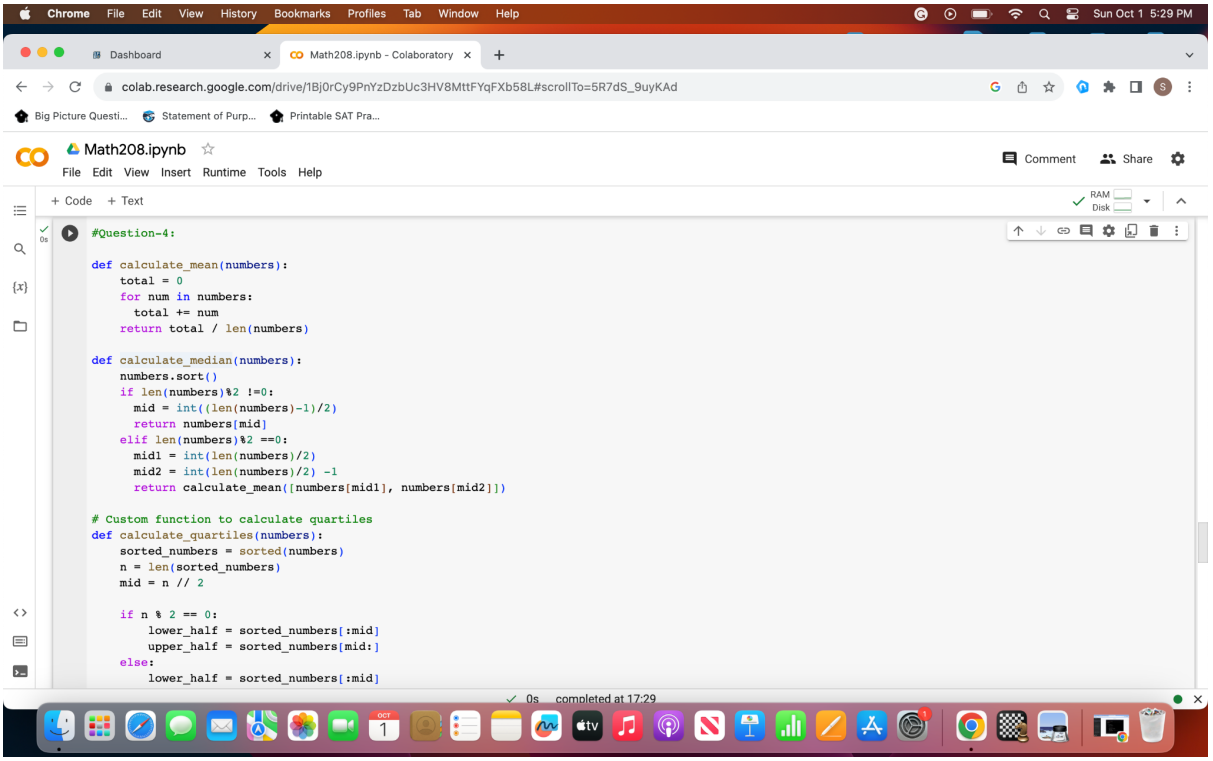
+ Code + Text

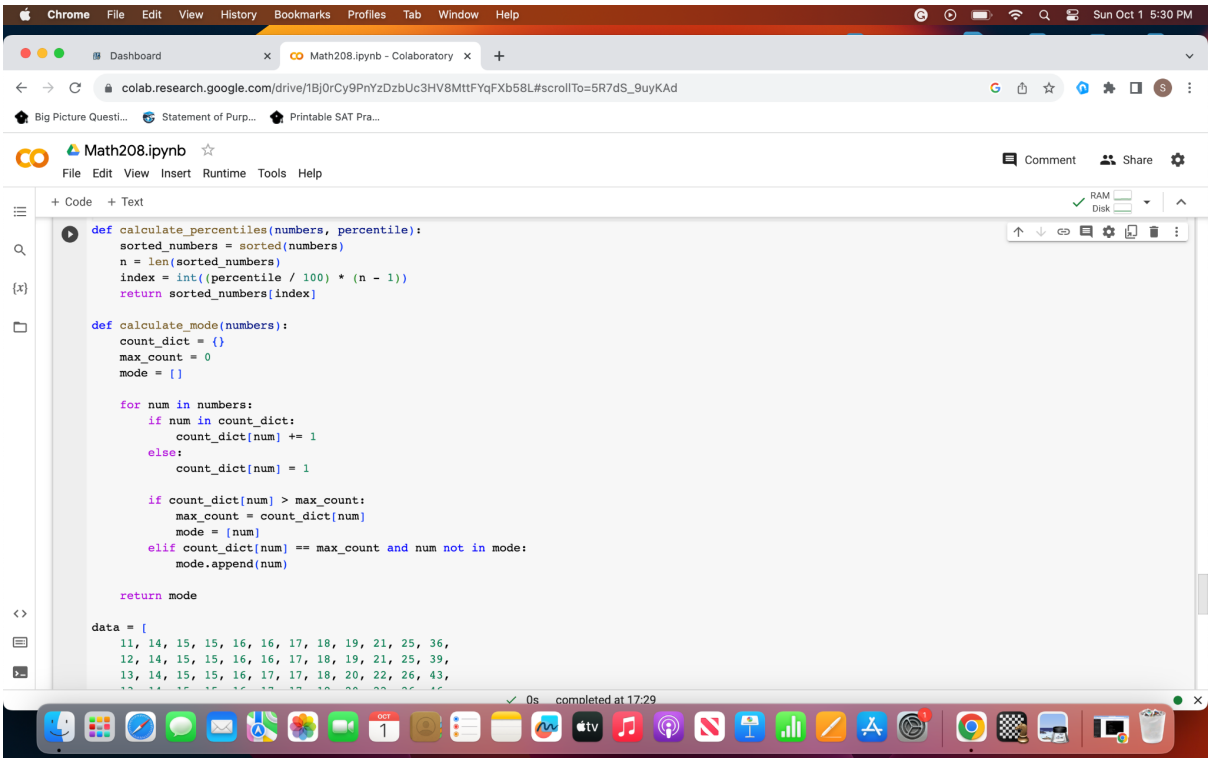
RAM Disk

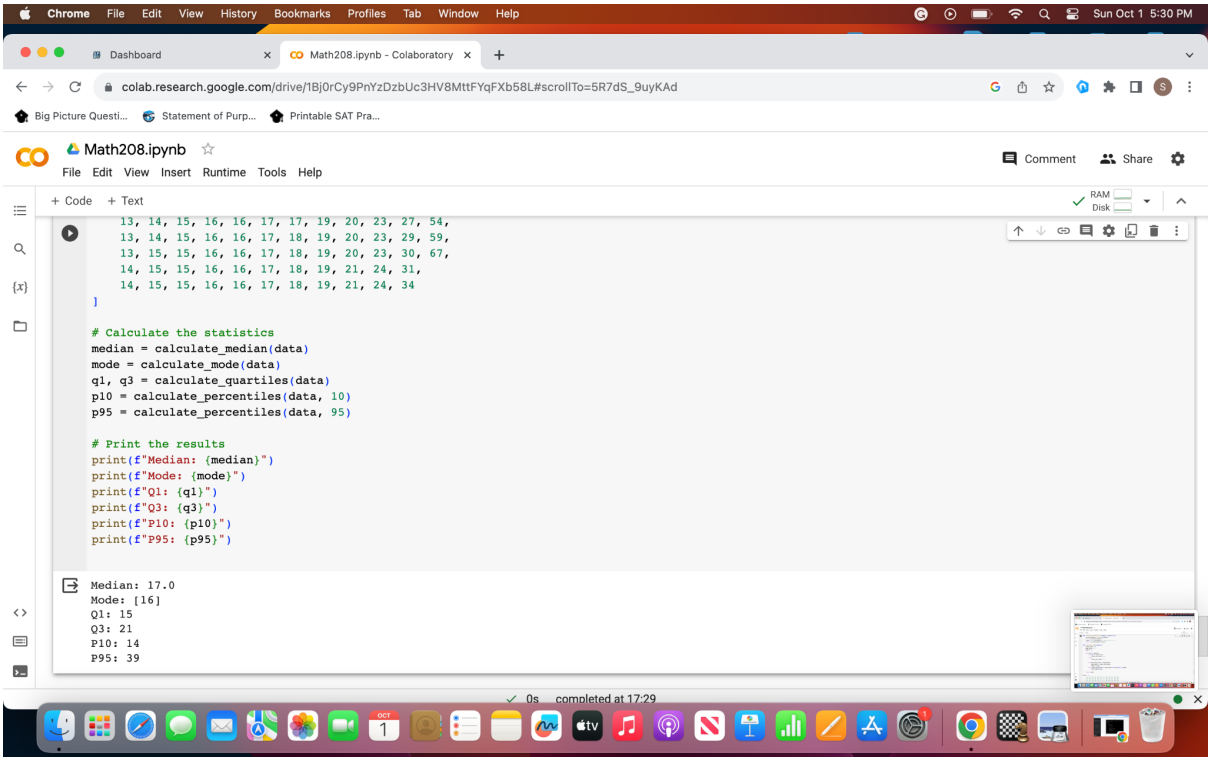
```
plt.tight_layout()
plt.show()
```



0s completed at 17:29


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun Oct 1 5:29 PM
Dashboard
Math208.ipynb - Colaboratory
colab.research.google.com/drive/1Bj0rCy9PnYzDzbUc3HV8MttFYqFXb58L#scrollTo=5R7dS_9uyKAd
Big Picture Questi...
Statement of Purp...
Printable SAT Pra...
Math208.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
#Question-4:
def calculate_mean(numbers):
total = 0
for num in numbers:
total += num
return total / len(numbers)
def calculate_median(numbers):
numbers.sort()
if len(numbers)%2 !=0:
mid = int((len(numbers)-1)/2)
return numbers[mid]
elif len(numbers)%2 ==0:
mid1 = int(len(numbers)/2)
mid2 = int(len(numbers)/2) -1
return calculate_mean([numbers[mid1], numbers[mid2]])
# Custom function to calculate quartiles
def calculate_quartiles(numbers):
sorted_numbers = sorted(numbers)
n = len(sorted_numbers)
mid = n // 2
if n % 2 == 0:
lower_half = sorted_numbers[:mid]
upper_half = sorted_numbers[mid:]
else:
lower_half = sorted_numbers[:mid]
0s completed at 17:29


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun Oct 1 5:30 PM
Dashboard
Math208.ipynb - Colaboratory
colab.research.google.com/drive/1Bj0rCy9PnYzDzbUc3HV8MttFYqFXb58L#scrollTo=5R7dS_9uyKAd
Big Picture Questi...
Statement of Purp...
Printable SAT Pra...
Math208.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
def calculate_percentiles(numbers, percentile):
sorted_numbers = sorted(numbers)
n = len(sorted_numbers)
index = int((percentile / 100) * (n - 1))
return sorted_numbers[index]
def calculate_mode(numbers):
count_dict = {}
max_count = 0
mode = []
for num in numbers:
if num in count_dict:
count_dict[num] += 1
else:
count_dict[num] = 1
if count_dict[num] > max_count:
max_count = count_dict[num]
mode = [num]
elif count_dict[num] == max_count and num not in mode:
mode.append(num)
return mode
data = [
11, 14, 15, 15, 16, 16, 17, 18, 19, 21, 25, 36,
12, 14, 15, 15, 16, 16, 17, 18, 19, 21, 25, 39,
13, 14, 15, 15, 16, 17, 17, 18, 20, 22, 26, 43,
0s completed at 17:29


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun Oct 1 5:30 PM
Dashboard
Math208.ipynb - Colaboratory
colab.research.google.com/drive/1Bj0rCy9PnYzDzbUc3HV8MttFYqFXb58L#scrollTo=5R7dS_9uyKAd
Big Picture Questi...
Statement of Purp...
Printable SAT Pra...
Math208.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code
+ Text
RAM
Disk
13, 14, 15, 16, 16, 17, 17, 19, 20, 23, 27, 54,
13, 14, 15, 16, 16, 17, 18, 19, 20, 23, 29, 59,
13, 15, 15, 16, 16, 17, 18, 19, 20, 23, 30, 67,
14, 15, 15, 16, 16, 17, 18, 19, 21, 24, 31,
14, 15, 15, 16, 16, 17, 18, 19, 21, 24, 34
]
# Calculate the statistics
median = calculate_median(data)
mode = calculate_mode(data)
q1, q3 = calculate_quartiles(data)
p10 = calculate_percentiles(data, 10)
p95 = calculate_percentiles(data, 95)
# Print the results
print(f"Median: {median}")
print(f"Mode: {mode}")
print(f"Q1: {q1}")
print(f"Q3: {q3}")
print(f"P10: {p10}")
print(f"P95: {p95}")
Median: 17.0
Mode: [16]
Q1: 15
Q3: 21
P10: 14
P95: 39
0s completed at 17:29