```r
library(dplyr)
library(moments)
library(lubridate)

analyze_time_series <- function(dataset) {
  dataset$Date <- as.Date(dataset$Date, format = "%d.%m.%Y")

  mean_value <- mean(dataset$Open)
  median_value <- median(dataset$Open)
  std_dev <- sd(dataset$Open)
  variance <- var(dataset$Open)

  skewness_kurtosis <- c(skewness(dataset$Open), kurtosis(dataset$Open))

  quartiles <- quantile(dataset$Open, c(0, 0.25, 0.5, 0.75, 1))

  correlation_coefficient <- cor(dataset$Open[-1], dataset$Open[-nrow(dataset)], method = "pearson")

  # Return results
  analysis_results <- list(mean = mean_value,
                           median = median_value,
                           std_deviation = std_dev,
                           variance = variance,
                           skewness_kurtosis = skewness_kurtosis,
                           quartiles = quartiles,
                           correlation_coefficient = correlation_coefficient)
  return(analysis_results)
}

time_series_data <- read.csv("/home/netherquark/Soham/Soham/Studies/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/R/beijing_multi_site_air_quality_data.csv", header = TRUE, stringsAsFactors = FALSE)
analysis_results <- analyze_time_series(time_series_data)
print(analysis_results)

# Function to visualize data
visualize_data <- function(dataset) {
  # Create scatter plot
  plot(dataset$Open, dataset$Volume, main = "Scatter Plot of Volume over Time", xlab = "Date", ylab = "Volume", col = "blue")

  # Create box plot
  boxplot(dataset$Open, main = "Box Plot of Volume", ylab = "Volume", col = "lightgreen")

  # Create histogram
  hist(dataset$Open, breaks = 20, main = "Histogram of Volume", xlab = "Volume", col = "skyblue", border = "black")
}

visualization_results <- visualize_data(time_series_data)
```

plot.png - STAIML-CA - VSCodium

File Edit Selection View Go Run Terminal Help

EXPLORER

- STAIML-CA
  - datasets
    - appliances_energ...
      - appliances_ener...
    - beijing_multi_site...
      - PRSA2017_Data_2...
      - beijing_multi_sit...
      - test.csv
    - fin_4400_pub
    - pharma_sales_data
    - advertisement_vs...
    - appliances-energ...
    - beijing-multi-site-...
    - fin_4400_pub.zip
    - pharma_sales_dat...
  - PowerBI
    - balanceSheetHisto...
    - q_3_a_pharma_s...
    - q3_b_fin_data.pbix
    - salesdaily.csv
    - STAIML-CA-2023-24...
  - Python
    - vscode
    - advertisements_v...
    - appliances_energ...
    - q2_a_moving_ave...
    - q2_b_compute_st...
    - q2_c_linear_regre...
    - q2_d_plotting.py
    - STAIML-CA-2023-24...
  - R
    - beijing_multi_site...
    - q1_a_ts_r_func.R
    - q1_b_dataframe.R
    - q1_c_loops_expla...
    - q1_d_ts_statistics...
    - STAIML-CA-2023-24...
  - LICENSE
  - README.md
  - STAIML-CA-2023-24.p...
  - STAIML-CA-2023-24z...

OUTLINE
TIMELINE
OPENCL DEVICES

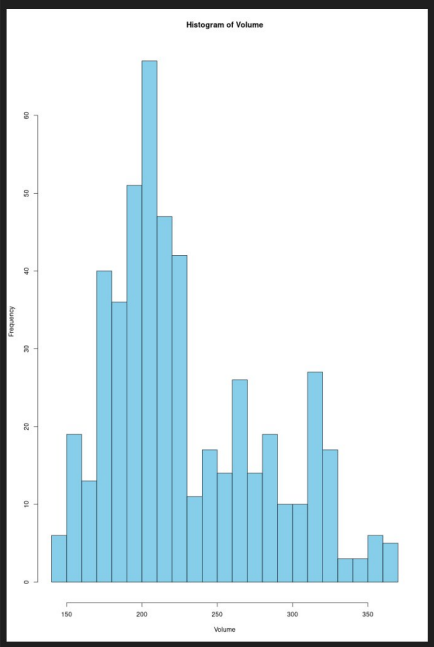
PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL PORTS

R interactive

```
> source("/home/netherquark/Soham/Soham/Studies/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/R/q1_a_ts_r_func.R", encoding = "UTF-8")
$mean
[1] 229.3914

$median
[1] 215.9

$std_deviation
[1] 51.69821

$variance
[1] 2672.705

$skewness_kurtosis
[1] 0.6986799 2.5692819

$quartiles
     0%     25%     50%     75%    100%
143.98 191.65 215.90 265.69 365.00

$correlation_coefficient
[1] 0.9950563

>
```

plot.png | q1_a_ts_r_func.R

tmp > RtmplJbXVj > vscode-R > plot.png



main 0 0 9 0

R 4.3.3: 131460 Whole Image 800x1200 16.04KB

```r
library(dplyr)
library(moments)
library(lubridate)

analyze_time_series ← function(dataset) {
  dataset$Date ← as.Date(dataset$Date, format = "%d.%m.%Y")

  mean_value ← mean(dataset$Open)
  median_value ← median(dataset$Open)
  std_dev ← sd(dataset$Open)
  variance ← var(dataset$Open)

  # Calculating skewness and kurtosis
  skewness_value ← skewness(dataset$Open)
  kurtosis_value ← kurtosis(dataset$Open)

  # Creating dataframe for quartiles
  quartiles_df ← data.frame(
    Quantile = c("0%", "25%", "50%", "75%", "100%"),
    Value = quantile(dataset$Open, c(0, 0.25, 0.5, 0.75, 1))
  )

  correlation_coefficient ← cor(dataset$Open[-1], dataset$Open[-nrow(dataset)], method = "pearson")

  # Creating dataframe for analysis results
  analysis_results ← data.frame(
    Attribute = c("Mean", "Median", "Standard Deviation", "Variance", "Skewness", "Kurtosis", "Correlation Coefficient"),
    Value = c(mean_value, median_value, std_dev, variance, skewness_value, kurtosis_value, correlation_coefficient)
  )

  return(analysis_results)
}

# Example usage
time_series_data ← read.csv("/home/netherquark/Soham/Soham/Studies/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/R/beijing_multi_site_air_quality_data.csv", header = TRUE, stringsAsFactors = FALSE)
analysis_results ← analyze_time_series(time_series_data)
print(analysis_results)
```

q1_b_dataframe.R - STAIML-CA - VSCodium

File Edit Selection View Go Run Terminal Help

EXPLORER

STAIML-CA
- datasets
  - appliances_energ...
  - appliances_ener...
  - beijing_multi_site...
    - PRSA2017_Data_2...
  - beijing_multi_sit...
  - test.csv
- fin_4400_pub
- pharma_sales_data
- advertisement_vs...
- appliances-energ...
- beijing-multi-site-...
- fin_4400_pub.zip
- pharma_sales_dat...
- PowerBI
  - balanceSheetHisto...
  - q_3_a_pharma_s...
  - q3_b_fin_data.pbix
  - salesdaily.csv
  - STAIML-CA-2023-24...
- Python
  - .vscode
  - advertisements_v...
  - appliances_energ...
  - q2_a_moving_ave...
  - q2_b_compute_st...
  - q2_c_linear_regre...
  - q2_d_plotting.py
  - STAIML-CA-2023-24...
- R
  - beijing_multi_site...
  - q1_a_ts_r_func.R
  - q1_b_datafra... !
  - q1_c_loops_expla...
  - q1_d_ts_statistics...
  - STAIML-CA-2023-24...
- LICENSE
- README.md
- STAIML-CA-2023-24.p...
- STAIML-CA-2023-24z...

OUTLINE
TIMELINE
OPENCL DEVICES

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

R interactive

```
> source("/home/netherquark/Soham/Soham/Studies/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/R/q1_b_dataframe.R", encoding = "UTF-8")
             Attribute       Value
1                 Mean  229.3914115
2               Median  215.8999940
3   Standard Deviation   51.6982146
4             Variance 2672.7053897
5             Skewness    0.6986799
6             Kurtosis    2.5692819
7 Correlation Coefficient    0.9950563
>
```

q1_b_dataframe.R ! ×

R > q1_b_dataframe.R > analyze_time_series

```r
library(dplyr)
library(moments)
library(lubridate)

analyze_time_series <- function(dataset) {
  dataset$Date <- as.Date(dataset$Date, format = "%
  mean_value <- mean(dataset$Open)
  median_value <- median(dataset$Open)
  std_dev <- sd(dataset$Open)
  variance <- var(dataset$Open)
  # Calculating skewness and kurtosis
  skewness_value <- skewness(dataset$Open)
  kurtosis_value <- kurtosis(dataset$Open)
  # Creating dataframe for quartiles
  quartiles_df <- data.frame(
    Quantile = c("0%", "25%", "50%", "75%", "100%"
    Value = quantile(dataset$Open, c(0, 0.25, 0.5,
  )
  correlation_coefficient <- cor(dataset$Open[-1],
  # Creating dataframe for analysis results
  analysis_results <- data.frame(
    Attribute = c("Mean", "Median", "Standard Devia
    Value = c(mean_value, median_value, std_dev, va
  )
  return(analysis_results)
}

# Example usage
time_series_data <- read.csv("/home/netherquark/Soh
analysis_results <- analyze_time_series(time_series
print(analysis_results)
```

main ⊗0 ⚠1 ⓘ10 0 R 4.3.3: 131460 Ln 23, Col 20 Spaces: 2 UTF-8 LF R

```r
# Function to generate Fibonacci sequence using for loop
fibonacci_for ← function(n) {
  fib ← numeric(n)
  fib[1] ← 0
  fib[2] ← 1
  for (i in 3:n) {
    cat("Loop number:", i)
    fib[i] ← fib[i - 1] + fib[i - 2]
    cat(" fib[i]:", fib[i], "\n")
  }
  return(fib)
}

# Function to generate Fibonacci sequence using while loop
fibonacci_while ← function(n) {
  fib ← numeric(n)
  fib[1] ← 0
  fib[2] ← 1
  i ← 3
  while (i ≤ n) {
    cat("Loop number:", i)
    fib[i] ← fib[i - 1] + fib[i - 2]
    cat(" fib[i]:", fib[i], "\n")
    i ← i + 1
  }
  return(fib)
}

# Print explanation for for loop
cat("Explanation (for loop):\n")
cat("The for loop in this program is used to generate the Fibonacci sequence. It starts by initializing an empty vector with 'numeric(n)' where 'n' is the desired length of the sequence. The first two elements of the sequence are set manually to 0 and 1. Then, the loop iterates from the third element onwards (i = 3 to n). At each iteration, the current Fibonacci number is calculated by adding the previous two numbers. Finally, the function returns the generated Fibonacci sequence.\n\n")
cat("Demonstration of Fibonacci sequence using for loop:\n \n")
# Print Fibonacci sequence generated using for loop
cat("\nFibonacci sequence using for loop: ", fibonacci_for(10), "\n")

# Print explanation for while loop
cat("\nExplanation for the while loop:\n")
cat("The while loop in this program is also used to generate the Fibonacci sequence. It starts by initializing an empty vector similar to the for loop method. A loop counter 'i' is initialized outside the loop and set to 3, the index of the third Fibonacci number. The loop continues until 'i' reaches 'n', the desired length of the sequence. Within the loop, the current Fibonacci number is calculated similar to the for loop method. The loop counter 'i' is then incremented by 1 in each iteration. Finally, the function returns the generated Fibonacci sequence.\n\n")
cat("Demonstration of Fibonacci sequence using for loop:\n \n")

# Print Fibonacci sequence generated using while loop
cat("\nFibonacci sequence using while loop: ", fibonacci_while(10), "\n")
```

q1_c_loops_explanation.R - STAIML-CA - VSCodium

File Edit Selection View Go Run Terminal Help

EXPLORER

STAIML-CA
- datasets
  - appliances_energ...
  - appliances_ener...
  - beijing_multi_site...
    - PRSA2017_Data_2...
  - beijing_multi_sit...
  - test.csv
  - fin_4400_pub
  - pharma_sales_data
  - advertisement_vs...
  - appliances-energ...
  - beijing-multi-site-...
  - fin_4400_pub.zip
  - pharma_sales_dat...
- PowerBI
  - balanceSheetHisto...
  - q_3_a_pharma_s...
  - q3_b_fin_data.pbix
  - salesdaily.csv
  - STAIML-CA-2023-24...
- Python
  - .vscode
  - advertisements_v...
  - appliances_energ...
  - q2_a_moving_ave...
  - q2_b_compute_st...
  - q2_c_linear_regre...
  - q2_d_plotting.py
  - STAIML-CA-2023-24...
- R
  - beijing_multi_site...
  - q1_a_ts_r_func.R
  - q1_b_dataframe.R
  - q1_c_loops_expla...
  - q1_d_ts_statistics...
  - STAIML-CA-2023-24...
- LICENSE
- README.md
- STAIML-CA-2023-24.p...
- STAIML-CA-2023-24z...

OUTLINE
TIMELINE
OPENCL DEVICES

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

R interactive

```
> source("/home/netherquark/Soham/Soham/Studies/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/R/q1_c_loops_explanation.R", encoding = "UTF-8")
Explanation (for loop):
The for loop in this program is used to generate the Fibonacci sequence. It starts by initializing an empty vector with 'numeric(n)' where 'n' is the desired length of the sequence. The first two elements of the sequence are set manually to 0 and 1. Then, the loop iterates from the third element onwards (i = 3 to n). At each iteration, the current Fibonacci number is calculated by adding the previous two numbers. Finally, the function returns the generated Fibonacci sequence.

Demonstration of Fibonacci sequence using for loop:

Loop number: 3 fib[i]: 1
Loop number: 4 fib[i]: 2
Loop number: 5 fib[i]: 3
Loop number: 6 fib[i]: 5
Loop number: 7 fib[i]: 8
Loop number: 8 fib[i]: 13
Loop number: 9 fib[i]: 21
Loop number: 10 fib[i]: 34

Fibonacci sequence using for loop:  0 1 1 2 3 5 8 13 21 34

Explanation for the while loop:
The while loop in this program is also used to generate the Fibonacci sequence. It starts by initializing an empty vector similar to the for loop method. A loop counter 'i' is initialized outside the loop and set to 3, the index of the third Fibonacci number. The loop continues until 'i' reaches 'n', the desired length of the sequence. Within the loop, the current Fibonacci number is calculated similar to the for loop method. The loop counter 'i' is then incremented by 1 in each iteration. Finally, the function returns the generated Fibonacci sequence.

Demonstration of Fibonacci sequence using for loop:

Loop number: 3 fib[i]: 1
Loop number: 4 fib[i]: 2
Loop number: 5 fib[i]: 3
Loop number: 6 fib[i]: 5
Loop number: 7 fib[i]: 8
Loop number: 8 fib[i]: 13
Loop number: 9 fib[i]: 21
Loop number: 10 fib[i]: 34

Fibonacci sequence using while loop:  0 1 1 2 3 5 8 13 21 34
>
```

q1_c_loops_explanation.R

R > q1_c_loops_explanation.R > ...

```r
# Function to generate Fibonacci sequence using for
fibonacci_for <- function(n) {
  fib <- numeric(n)
  fib[1] <- 0
  fib[2] <- 1
  for (i in 3:n) {
    cat("Loop number:", i)
    fib[i] <- fib[i - 1] + fib[i - 2]
    cat(" fib[i]:", fib[i], "\n")
  }
  return(fib)
}

# Function to generate Fibonacci sequence using wh
fibonacci_while <- function(n) {
  fib <- numeric(n)
  fib[1] <- 0
  fib[2] <- 1
  i <- 3
  while (i <= n) {
    cat("Loop number:", i)
    fib[i] <- fib[i - 1] + fib[i - 2]
    cat(" fib[i]:", fib[i], "\n")
    i <- i + 1
  }
  return(fib)
}

# Print explanation for for loop
cat("Explanation (for loop):\n")
cat("The for loop in this program is used to genera
cat("Demonstration of Fibonacci sequence using for
# Print Fibonacci sequence generated using for loo
cat("\nFibonacci sequence using for loop: ", fibona

# Print explanation for while loop
cat("\nExplanation for the while loop:\n")
cat("The while loop in this program is also used to
cat("Demonstration of Fibonacci sequence using for

# Print Fibonacci sequence generated using while lo
cat("\nFibonacci sequence using while loop: ", fibo
```

main ⊗0 ⚠0 ⓘ2 0 R 4.3.3: 131460 Ln 1, Col 1 Spaces: 2 UTF-8 LF R

```r
analyze_correlation <- function(dataset) {
  correlation_coefficient <- cor(dataset$Open, dataset$High, method = "pearson")

  # Determine statistical significance
  n <- nrow(dataset)
  degrees_freedom <- n - 2
  t_value <- abs(correlation_coefficient) * sqrt(degrees_freedom) / sqrt(1 - correlation_coefficient^2)
  p_value <- 2 * pt(-abs(t_value), df = degrees_freedom)
  is_significant <- p_value < 0.05

  analysis_results <- list(
    correlation_coefficient = correlation_coefficient,
    is_significant = is_significant,
    p_value = p_value
  )
  return(analysis_results)
}

time_series_data <- read.csv("/home/netherquark/Soham/Soham/Studies/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/R/beijing_multi_site_air_quality_data.csv", header = TRUE, stringsAsFactors = FALSE)

analysis_results <- analyze_correlation(time_series_data)
print(analysis_results)

visualize_data <- function(dataset) {
  plot(dataset$Open, dataset$High, main = "Scatter Plot of Open vs High", xlab = "Open", ylab = "High", col = "blue")
}

visualize_data(time_series_data)
```

plot.png - STAIML-CA - VSCodium

File Edit Selection View Go Run Terminal Help

EXPLORER

STAIML-CA
- datasets
  - appliances_energ...
  - appliances_ener...
  - beijing_multi_site...
    - PRSA2017_Data_2...
  - beijing_multi_sit...
  - test.csv
  - fin_4400_pub
  - pharma_sales_data
  - advertisement_vs...
  - appliances-energ...
  - beijing-multi-site-...
  - fin_4400_pub.zip
  - pharma_sales_dat...
- PowerBI
  - balanceSheetHisto...
  - q_3_a_pharma_s...
  - q3_b_fin_data.pbix
  - salesdaily.csv
  - STAIML-CA-2023-24...
- Python
  - .vscode
  - advertisements_v...
  - appliances_energ...
  - q2_a_moving_ave...
  - q2_b_compute_st...
  - q2_c_linear_regre...
  - q2_d_plotting.py
  - STAIML-CA-2023-24...
- R
  - beijing_multi_site...
  - q1_a_ts_r_func.R
  - q1_b_dataframe.R
  - q1_c_loops_expla...
  - q1_d_ts_statistics...
  - STAIML-CA-2023-24...
- LICENSE
- README.md
- STAIML-CA-2023-24.p...
- STAIML-CA-2023-24z...

OUTLINE
TIMELINE
OPENCL DEVICES

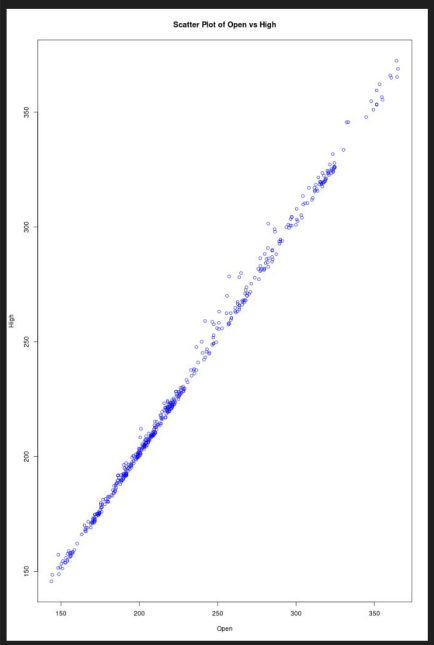
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS

R interactive

```
> source("/home/netherquark/Soham/Soham/Studies/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/R/q1_d_ts_statistics.R", encoding = "UTF-8")
$correlation_coefficient
[1] 0.9987594

$is_significant
[1] TRUE

$p_value
[1] 0

>
```

plot.png | q1_d_ts_statistics.R

tmp > RtmpUJbXVj > vscode-R > plot.png



main 0 0 5 0

R 4.3.3: 131460 Whole Image 800x1200 48.36KB

```python
import numpy as np

class MovingAverageTS:

    def __init__(self, window):
        self.window = window
        self.time_series_sequence = np.random.rand(100)

    def moving_average(self):
        weights = (np.repeat(1.0, self.window)) / self.window
        return np.convolve(self.time_series_sequence, weights, 'valid')

window = 10

moving_average_inst = MovingAverageTS(window)
result = moving_average_inst.moving_average()
print("Moving averages are: \n", result)
```

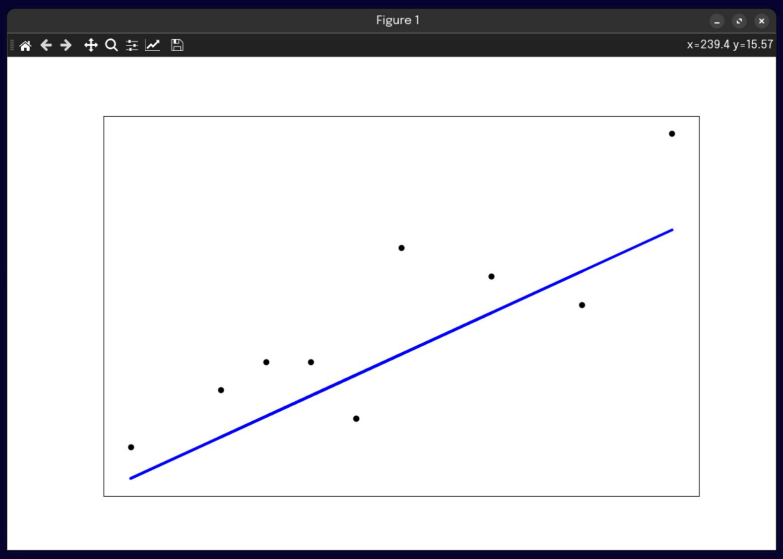
```
~/So/S/s/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/Python on  main at 05:25:14 PM
python3 q2_a_moving_average_ts.py
Moving averages are:
 [0.46860009 0.54235564 0.48650917 0.42587579 0.47174599 0.44044231
0.45918165 0.40320774 0.40165814 0.379087   0.431095   0.39474589
0.40259913 0.4145805  0.36414114 0.37031703 0.3339693  0.39983078
0.45877355 0.46238158 0.4734909  0.51495776 0.51181965 0.48109444
0.46776298 0.50317737 0.51801594 0.4671704  0.47807757 0.48592204
0.45369273 0.37961426 0.4154181  0.44029552 0.41469228 0.40999768
0.39493177 0.42339052 0.40719264 0.42182331 0.40046733 0.48155646
0.42915884 0.45916861 0.52116795 0.5647194  0.62727424 0.61305205
0.60556776 0.5617643  0.62341753 0.58977642 0.65588295 0.60604179
0.55318706 0.5232941  0.52862616 0.50171061 0.49614505 0.46893432
0.42742648 0.4141088  0.40284889 0.46093082 0.48030393 0.42832261
0.37723748 0.36547115 0.3399947  0.39288969 0.41469365 0.37460978
0.35565216 0.35460764 0.39876843 0.40449755 0.43936947 0.46026395
0.45902824 0.49861371 0.47663461 0.51031814 0.47981097 0.42659012
0.43089996 0.4989655  0.47429148 0.43724552 0.48627415 0.43268277
0.40533084]
~/So/S/s/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/Python on  main at 05:25:31 PM
```

```python
import pandas as pd

df_appliances = pd.read_csv('appliances_energy_prediction.csv')
print(df_appliances.describe())
```

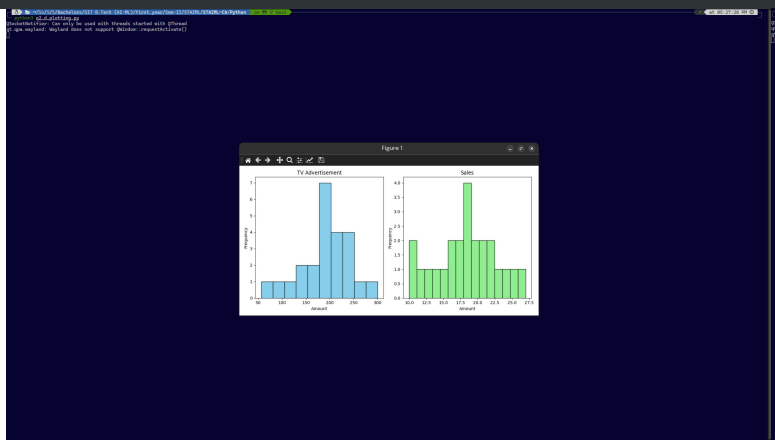
```
~/So/S/s/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/Python on  main  at 05:26:00 PM
python3 q2_b_compute_stats.py
         Appliances        lights            T1         RH_1            T2         RH_2            T3         RH_3            T4  ...         RH_9         T_out  Press_mm_hg       RH_out    Windspeed    Visibility     Tdewpoint          rv1          rv2
count  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000  ...  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000
mean      97.694958      3.801875     21.686571     40.259739     20.341219     40.420420     22.267611     39.242500     20.855335  ...     41.552401      7.411665    755.522602     79.750418      4.039752     38.330834      3.760707     24.988033     24.988033
std      102.524891      7.935988      1.606066      3.979299      2.192974      4.069813      2.006111      3.254576      2.042884  ...      4.151497      5.317409      7.399441     14.901088      2.451221     11.794719      4.194648     14.496634     14.496634
min       10.000000      0.000000     16.790000     27.023333     16.100000     20.463333     17.200000     28.766667     15.100000  ...     29.166667     -5.000000    729.300000     24.000000      0.000000      1.000000     -6.600000      0.005322      0.005322
25%       50.000000      0.000000     20.760000     37.333333     18.790000     37.900000     20.790000     36.900000     19.530000  ...     38.500000      3.666667    750.933333     70.333333      2.000000     29.000000      0.900000     12.497889     12.497889
50%       60.000000      0.000000     21.600000     39.656667     20.000000     40.500000     22.100000     38.530000     20.666667  ...     40.900000      6.916667    756.100000     83.666667      3.666667     40.000000      3.433333     24.897653     24.897653
75%      100.000000      0.000000     22.600000     43.066667     21.500000     43.260000     23.290000     41.760000     22.100000  ...     44.338095     10.408333    760.933333     91.666667      5.500000     40.000000      6.566667     37.583769     37.583769
max     1080.000000     70.000000     26.260000     63.360000     29.856667     56.026667     29.236000     50.163333     26.200000  ...     53.326667     26.100000    772.300000    100.000000     14.000000     66.000000     15.500000     49.996530     49.996530

[8 rows x 28 columns]
~/So/S/s/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/Python on  main  at 05:26:09 PM
```

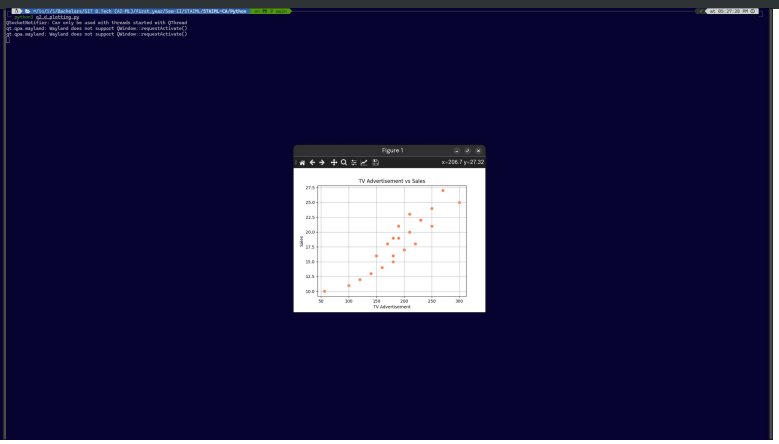
```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score

advertisements_vs_sales = pd.read_csv('advertisements_vs_sales.csv')

tv_advertisements = advertisements_vs_sales['TV Advertisement (X₁)'].values
sales = advertisements_vs_sales['Sales(Y)'].values

tv_advertisements_train = tv_advertisements[:-10].reshape(-1, 1)
tv_advertisements_test = tv_advertisements[-10:].reshape(-1, 1)

sales_train = sales[:-10]
sales_test = sales[-10:]

linregmodel = linear_model.LinearRegression()

linregmodel.fit(tv_advertisements_train, sales_train)
sales_predict = linregmodel.predict(tv_advertisements_test)

print("Coefficient: \n", linregmodel.coef_)
print("Mean squared error: %.2f" % mean_squared_error(sales_test, sales_predict))
print("Coefficient of determination: %.2f" % r2_score(sales_test, sales_predict))

plt.scatter(tv_advertisements_test, sales_test, color="black")
plt.plot(tv_advertisements_test, sales_predict, color="blue", linewidth=3)

plt.xticks(())
plt.yticks(())

plt.show()
```

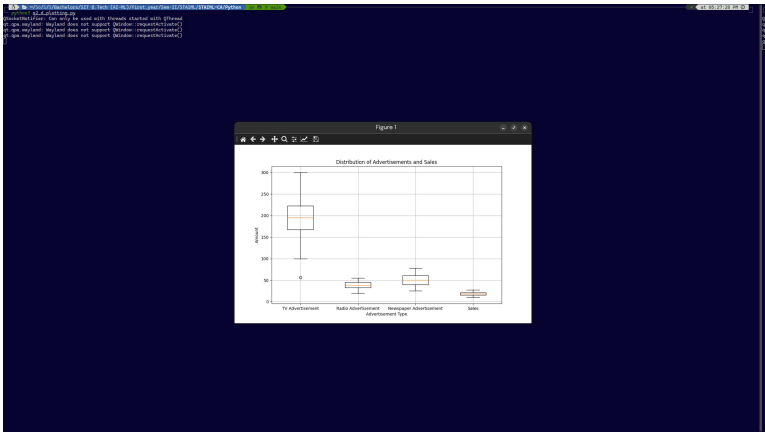
```
~/So/S/s/Bachelors/SIT B.Tech (AI-ML)/first_year/Sem-II/STAIML/STAIML-CA/Python on  main                    at 05:26:37 PM
 python3 q2_c_linear_regression.py
Coefficient:
 [0.07263148]
Mean squared error: 4.22
Coefficient of determination: 0.59
QSocketNotifier: Can only be used with threads started with QThread
qt.qpa.wayland: Wayland does not support QWindow::requestActivate()
```

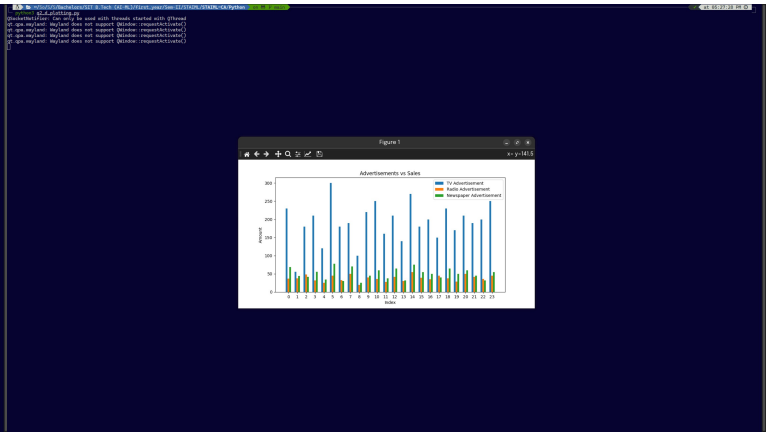


```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score

advertisements_vs_sales = pd.read_csv('advertisements_vs_sales.csv')

tv_advertisements = advertisements_vs_sales['TV Advertisement (X₁)'].values
sales = advertisements_vs_sales['Sales(Y)'].values

tv_advertisements_train = tv_advertisements[:-10].reshape(-1, 1)
tv_advertisements_test = tv_advertisements[-10:].reshape(-1, 1)

sales_train = sales[:-10]
sales_test = sales[-10:]

linregmodel = linear_model.LinearRegression()

linregmodel.fit(tv_advertisements_train, sales_train)
sales_predict = linregmodel.predict(tv_advertisements_test)

print("Coefficient: \n", linregmodel.coef_)
print("Mean squared error: %.2f" % mean_squared_error(sales_test, sales_predict))
print("Coefficient of determination: %.2f" % r2_score(sales_test, sales_predict))

plt.scatter(tv_advertisements_test, sales_test, color="black")
plt.plot(tv_advertisements_test, sales_predict, color="blue", linewidth=3)

plt.xticks(())
plt.yticks(())

plt.show()
```





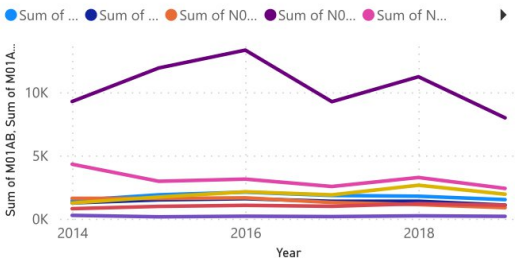




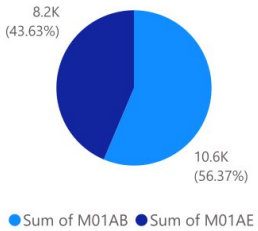
List of sales

10,600.94 Sum of M01AB	8,204.62 Sum of M01AE
8,172.21 Sum of N02BA	63,005.40 Sum of N02BE
18646 Sum of N05B	1248 Sum of N05C
11607 Sum of R03	6,107.82 Sum of R06

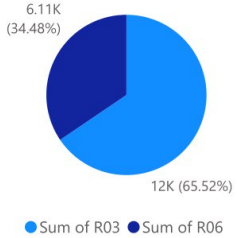
Sales over time



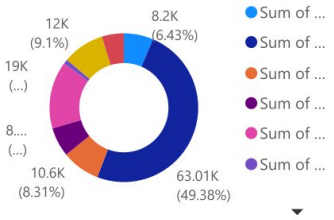
M01Ax split



R0x split



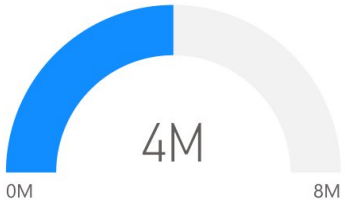
Division of sales



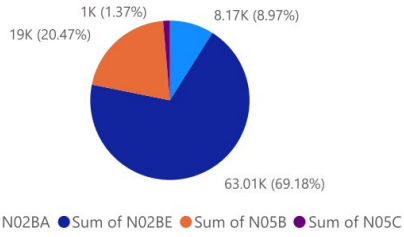
Year, Month

- 2014
- 2015
- 2016
- 2017
- 2018
- 2019

Annual sale



N0xxx split

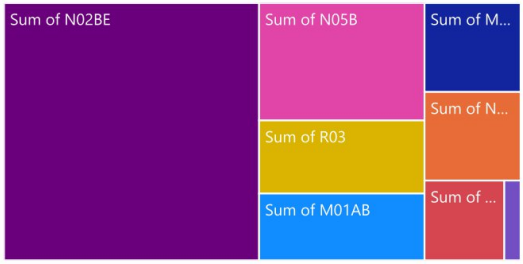


Ask a question about your data

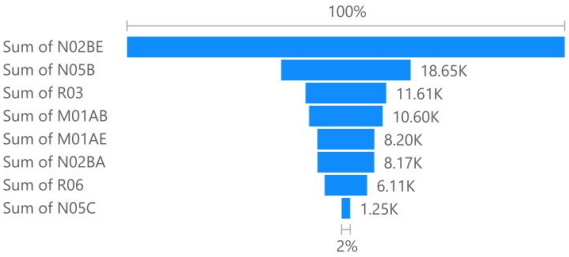
Try one of these to get started

- total hour
- average r 06
- count weekday names

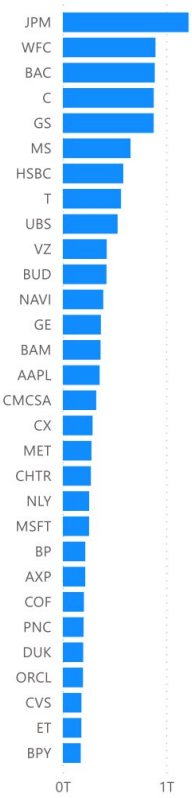
Split of pharmaceutical sales



Ranked sales



Sum of longTermDebt by stock



Sum of longTermDebt by stock

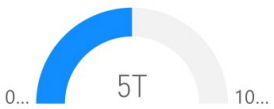
Sum of cash

12T

Sum of longTermDebt

39T

Sum of inventory

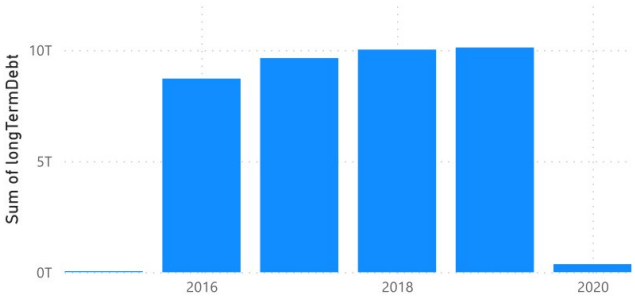


totalAssets: 82 – 2715152000000

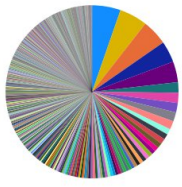
Year

- 2012
- 2013
- 2014

Sum of longTermDebt by Year



Sum of totalAssets by stock



stock
- JPM
- HSBC
- BAC
- WFC
- C
- UBS
- GS
- MS
- PRU

Ask a question about your data

Try one of these to get started

- count stocks
- total inventory
- average retained earnings