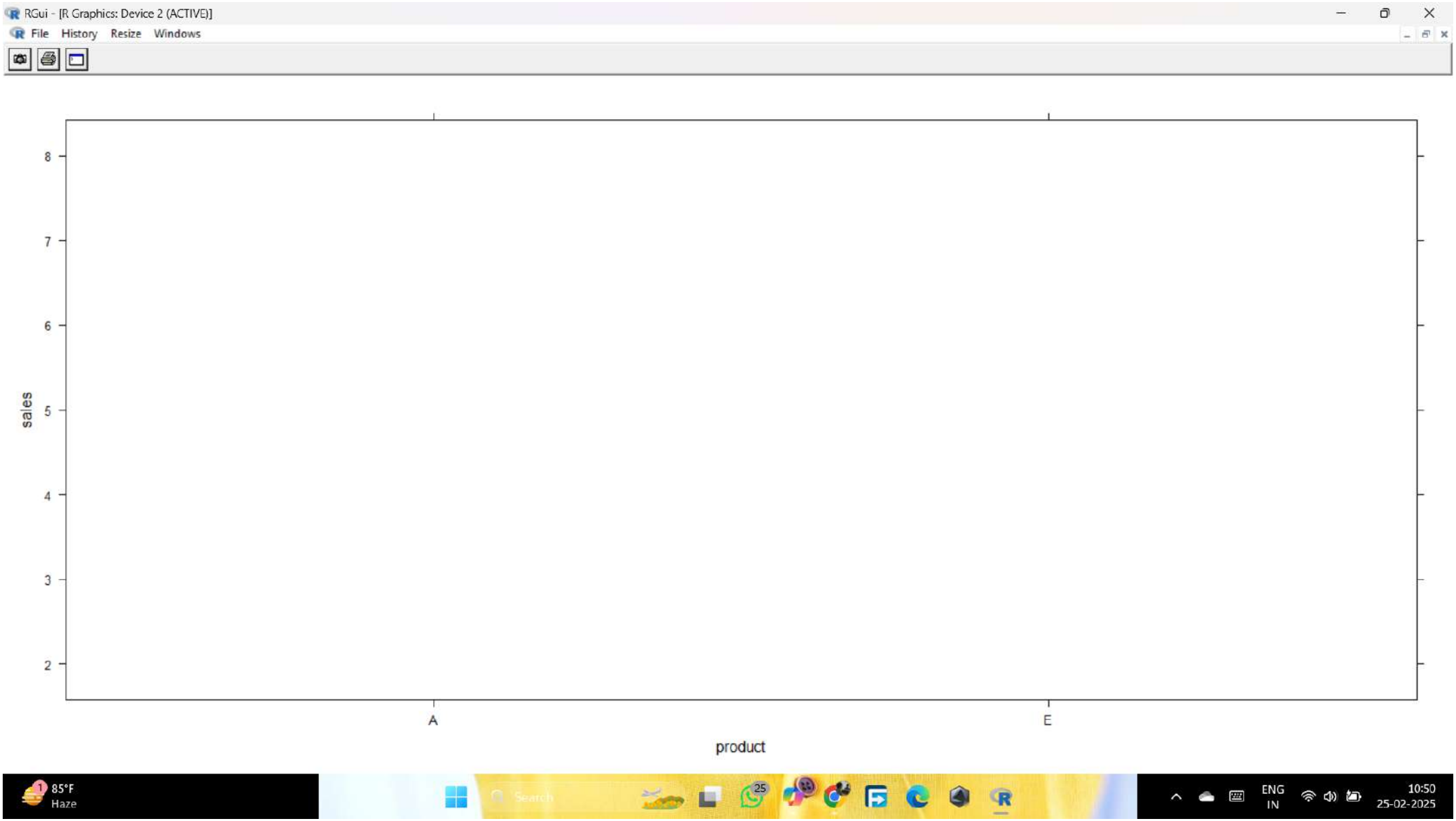
```
>
> # Example salary data
> salaries <- c(50000, 55000, 60000, 65000, 70000, 75000, 80000, 85000, 90000, 95000)
>
> # Calculate mean, median, and mode
> mean_salary <- mean(salaries)
> median_salary <- median(salaries)
> mode_salary <- as.numeric(names(sort(table(salaries), decreasing = TRUE)[1]))
>
> # Print results
> cat("Mean Salary:", mean_salary, "\n")
Mean Salary: 72500
> cat("Median Salary:", median_salary, "\n")
Median Salary: 72500
> cat("Mode Salary:", mode_salary, "\n")
Mode Salary: 50000
>
```

RGui - [R Console]

File Edit View Misc Packages Windows Help

```
> # Create the dataset
> Product_Sales <- data.frame(
+   product = c("A", "B", "C", "D", "E"),
+   sales = c(3, 5, 2, 6, 7)
+ )
>
> # Perform Poisson regression
> model <- glm(sales ~ product, data = Product_Sales, family = poisson())
>
> # Create scatter plot with regression line
> plot(Product_Sales$product, Product_Sales$sales)
Error in plot.window(...) : need finite 'xlim' values
In addition: Warning messages:
1: In xy.coords(x, y, xlabel, ylabel, log) : NAs introduced by coercion
2: In min(x) : no non-missing arguments to min; returning Inf
3: In max(x) : no non-missing arguments to max; returning -Inf
> abline(model, col = "red")
Warning message:
In abline(model, col = "red") :
  only using the first two of 5 regression coefficients
>
> # Evaluate model accuracy
> summary(model)

Call:
glm(formula = sales ~ product, family = poisson(), data = Product_Sales)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.0986     0.5774   1.903   0.0571 .
productB      0.5108     0.7303   0.699   0.4843
productC     -0.4055     0.9129  -0.444   0.6569
productD      0.9808     0.6770   1.449   0.1474
productE      0.8473     0.6901   1.228   0.2195
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 5.5006e+00  on 4  degrees of freedom
Residual deviance: 2.2204e-16  on 0  degrees of freedom
AIC: 26.832

Number of Fisher Scoring iterations: 3
```


RGui - [R Graphics: Device 2 (ACTIVE)]
File History Resize Windows
sales
8
7
6
5
4
3
2
A
E
product
85°F
Haze
Search
ENG
IN
10:50
25-02-2025

RGui - [R Console]

File Edit View Misc Packages Windows Help

```
>
> # Create the dataset
> Product_Sales <- data.frame(
+   product = c("A", "B", "C", "D", "E"),
+   sales = c(3, 5, 2, 9, 7)
+ )
>
> # Perform Poisson regression
> model <- glm(sales ~ product, data = Product_Sales, family = poisson())
>
> # Create scatter plot with regression line
> plot(Product_Sales$product, Product_Sales$sales)
Error in plot.window(...) : need finite 'xlim' values
In addition: Warning messages:
1: In xy.coords(x, y, xlabel, ylabel, log) : NAs introduced by coercion
2: In min(x) : no non-missing arguments to min; returning Inf
3: In max(x) : no non-missing arguments to max; returning -Inf
> abline(model, col = "red")
Warning message:
In abline(model, col = "red") :
  only using the first two of 5 regression coefficients
>
> # Evaluate model accuracy
> summary(model)

Call:
glm(formula = sales ~ product, family = poisson(), data = Product_Sales)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.0986     0.5774   1.903   0.0571 .
productB      0.5108     0.7303   0.699   0.4843
productC     -0.4055     0.9129  -0.444   0.6569
productD      0.9808     0.6770   1.449   0.1474
productE      0.8473     0.6901   1.228   0.2195
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 5.5006e+00  on 4  degrees of freedom
Residual deviance: 2.2204e-16  on 0  degrees of freedom
AIC: 26.832
```

```
>
> # Vectors of scores for each subject
> math_scores <- c(85, 90, 78, 92, 88)
> science_scores <- c(80, 85, 82, 90, 87)
> english_scores <- c(88, 91, 85, 89, 84)
>
> # Combine vectors into a matrix
> student_scores <- cbind(math_scores, science_scores, english_scores)
>
> # Calculate the average score for each student
> average_scores <- rowMeans(student_scores)
>
> # Print the matrix and average scores
> print(student_scores)
     math_scores science_scores english_scores
[1,]          85             80             88
[2,]          90             85             91
[3,]          78             82             85
[4,]          92             90             89
[5,]          88             87             84
> print(average_scores)
[1] 84.33333 88.66667 81.66667 90.33333 86.33333
>
```

```
> # Example wide-format dataset
> data <- data.frame(
+   product = c("A", "B", "C"),
+   Jan = c(100, 150, 200),
+   Feb = c(120, 160, 210),
+   Mar = c(130, 170, 220)
+ )
>
> # Melting the dataset to long-format
> long_data <- melt(data, id.vars = "product", variable.name = "month", value.name = "sales")
Error in melt(data, id.vars = "product", variable.name = "month", value.name = "sales") :
  could not find function "melt"
>
> # Casting the dataset back to wide-format
> wide_data <- dcast(long_data, product ~ month, value.var = "sales")
Error in dcast(long_data, product ~ month, value.var = "sales") :
  could not find function "dcast"
>
> # Print results
> print(long_data)
Error: object 'long_data' not found
> print(wide_data)
Error: object 'wide_data' not found
>
```

```r
# Create the data frame
data <- data.frame(
  Month = 1:6,
  Spends = c(1000, 4000, 5000, 4500, 3000, 4000),
  Sales = c(9914, 40487, 54324, 50044, 34719, 42551)
)

# Create a regression model
model <- lm(Sales ~ Spends, data = data)

# Print summary of the model
summary(model)
```

```
all:
m(formula = Sales ~ Spends, data = data)

esiduals:
     1       2       3       4       5       6
-525.7 -2740.0   167.9  1352.5  2421.1  -676.0

oefficients:
            Estimate Std. Error t value Pr(>|t|)
Intercept) -489.4490  2384.0522  -0.205    0.847
pends        10.9291     0.6252  17.481 6.29e-05 ***
--
ignif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

esidual standard error: 1997 on 4 degrees of freedom
ultiple R-squared:  0.9871,    Adjusted R-squared:  0.9839
-statistic: 305.6 on 1 and 4 DF,  p-value: 6.287e-05
```

```
>
> # Temperature data
> City_A <- c(22, 25, 21, 23, 24, 22, 20)
> City_B <- c(18, 20, 19, 21, 20, 19, 18)
> City_C <- c(30, 32, 31, 33, 34, 32, 31)
>
> # Calculate average temperature for each city
> avg_temp_A <- mean(City_A)
> avg_temp_B <- mean(City_B)
> avg_temp_C <- mean(City_C)
>
> # Identify the city with highest average temperature
> max_avg_temp <- max(avg_temp_A, avg_temp_B, avg_temp_C)
> city_with_max_temp <- ifelse(max_avg_temp == avg_temp_A, "City A", ifelse(max_avg_temp == avg_temp_B, "City B", "City C"))
>
> # Calculate variance in temperature for each city
> var_temp_A <- var(City_A)
> var_temp_B <- var(City_B)
> var_temp_C <- var(City_C)
>
> # Print results
> cat("Average Temperature for City A:", avg_temp_A, "\n")
Average Temperature for City A: 22.42857
> cat("Average Temperature for City B:", avg_temp_B, "\n")
Average Temperature for City B: 19.28571
> cat("Average Temperature for City C:", avg_temp_C, "\n")
Average Temperature for City C: 31.85714
> cat("City with highest average temperature:", city_with_max_temp, "with", max_avg_temp, "degrees Celsius\n")
City with highest average temperature: City C with 31.85714 degrees Celsius
> cat("Variance in temperature for City A:", var_temp_A, "\n")
Variance in temperature for City A: 2.952381
> cat("Variance in temperature for City B:", var_temp_B, "\n")
Variance in temperature for City B: 1.238095
> cat("Variance in temperature for City C:", var_temp_C, "\n")
Variance in temperature for City C: 1.809524
>
```

```
>
> # Example salary data
> salaries <- c(50000, 55000, 60000, 65000, 70000, 75000, 80000, 85000, 90000, 95000)
>
> # Calculate mean, median, and mode
> mean_salary <- mean(salaries)
> median_salary <- median(salaries)
> mode_salary <- as.numeric(names(sort(table(salaries), decreasing = TRUE)[1]))
>
> # Print results
> cat("Mean Salary:", mean_salary, "\n")
Mean Salary: 72500
> cat("Median Salary:", median_salary, "\n")
Median Salary: 72500
> cat("Mode Salary:", mode_salary, "\n")
Mode Salary: 50000
>
```

RGui - [R Console]

File Edit View Misc Packages Windows Help

```
>
> # Create data frame
> df <- data.frame(
+   ID = 1:5,
+   NAME = c("Alice", "Bob", "Carol", "Dave", "Eve"),
+   OCCUPATION = c("Engineer", "Doctor", "Teacher", "Artist", "Lawyer"),
+   EMPLOYEE = TRUE
+ )
>
> # Print structure of the data frame
> str(df)
'data.frame':   5 obs. of  4 variables:
 $ ID        : int  1 2 3 4 5
 $ NAME      : chr  "Alice" "Bob" "Carol" "Dave" ...
 $ OCCUPATION: chr  "Engineer" "Doctor" "Teacher" "Artist" ...
 $ EMPLOYEE  : logi  TRUE TRUE TRUE TRUE TRUE
>
> # Apply summary() to find summary of the data in the data frame
> summary(df)
       ID        NAME            OCCUPATION        EMPLOYEE
 Min.   :1   Length:5           Length:5           Mode:logical
 1st Qu.:2   Class :character   Class :character   TRUE:5
 Median :3   Mode  :character   Mode  :character
 Mean   :3
 3rd Qu.:4
 Max.   :5
>
> # Extract OCCUPATION column
> occupations <- df$OCCUPATION
>
> # Expand data frame including SALARY
> df$SALARY <- c(50000, 60000, 55000, 62000, 58000)
>
> # Print results
> print(df)
  ID  NAME OCCUPATION EMPLOYEE SALARY
1  1 Alice   Engineer     TRUE  50000
2  2   Bob     Doctor     TRUE  60000
3  3 Carol    Teacher     TRUE  55000
4  4  Dave     Artist     TRUE  62000
5  5   Eve     Lawyer     TRUE  58000
> print(occupations)
[1] "Engineer" "Doctor"   "Teacher"  "Artist"   "Lawyer"
>
```

RGui - [R Console]

File Edit View Misc Packages Windows Help

```
>
>
>
>
> # Read the input.csv file
> data <- read.csv("input.csv")
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file 'input.csv': No such file or directory
>
> # Get the maximum salary
> max_salary <- max(data$salary)
Warning message:
In max(data$salary) : no non-missing arguments to max; returning -Inf
>
> # Get details of the person with max salary
> person_max_salary <- data[which.max(data$salary), ]
>
> # Get all people working in IT department
> it_department <- subset(data, department == "IT")
Error in eval(e, x, parent.frame()) : object 'department' not found
>
> # Get the persons in IT department whose salary is greater than 600
> it_high_salary <- subset(it_department, salary > 600)
Error: object 'it_department' not found
>
> # Get people who joined on or after 2014
> joined_after_2014 <- subset(data, as.Date(join_date) >= as.Date("2014-01-01"))
Error in eval(e, x, parent.frame()) : object 'join_date' not found
>
> # Print results
> cat("Maximum Salary:", max_salary, "\n")
Maximum Salary: -Inf
> cat("Person with Max Salary:", person_max_salary, "\n")
Person with Max Salary: Error in cat("Person with Max Salary:", person_max_salary, "\n") :
  argument 2 (type 'list') cannot be handled by 'cat'
> cat("People in IT Department:", it_department, "\n")
Error: object 'it_department' not found
> cat("IT Department with Salary > 600:", it_high_salary, "\n")
Error: object 'it_high_salary' not found
> cat("People who Joined on or After 2014:", joined_after_2014, "\n")
Error: object 'joined_after_2014' not found
>
```