

MVLU COLLEGE

Subject:-Data Analysis with SAS / SPSS /R

7 Performing one-way ANOVA using aov() (R).

The screenshot shows the RStudio interface with the following details:

- File Menu:** File Edit Code View Plots Session Build Debug Profile Tools Help
- Project:** Project (None)
- Code Editor:** Shows R script code for reading a CSV file, checking column names, converting categorical variables to factors, applying one-way ANOVA, and displaying the results.
- Console:** Displays the R session output, including the command history and the ANOVA summary table.
- Environment:** Shows the global environment with objects like anova_result, boston, CarPrice_Ass..., chi_result, data, dep_anx_subs..., female, den_f, and megaGymDataset.
- Files:** Shows a list of files in the current directory, including Sleep_Efficiency (1).csv, Student Mental health.csv, student_exam_scores.csv, StudentsPerformance.csv, test.csv, Untitled.R, WA_Fn-UseC_-Telco-Customer-Churn.csv, xAPI-Edu-Data (1).csv, S125 M2 P7.R, S125 M2 P8.R, S125 M2 P9.R, Heart_Disease_Prediction.csv, indian_weather_data.csv, and megaGymDataset.csv.

```
R Studio
File Edit Code View Plots Session Build Debug Profile Tools Help
Project: (None)
S125 M2 P7.R | Heart_Disease_Prediction | S125 M2 P8.R | indian_weather_data | S125 M2 P9.R | megaGymDataset
Source on Save | Go to file/function | Addins |
R Script
4 data <- read.csv("Heart_Disease_Prediction.csv")
5
6 # Check column names
7 colnames(data)
8
9 # Convert categorical variable to factor
10 data$Sex <- as.factor(data$Sex)
11
12 # Apply One-way ANOVA
13 anova_result <- aov(Cholesterol ~ Sex, data = data)
14
15 # Display ANOVA table
17:1 [Top Level] c
R 4.5.2 . ~/~>
> # Load dataset
> data <- read.csv("Heart_Disease_Prediction.csv")
> # Check column names
> colnames(data)
[1] "Age"                  "Sex"                   "Chest.pain.type"
[4] "BP"                    "Cholesterol"          "FBS.over.120"
[7] "EKG.results"          "Max.HR"                "Exercise.angina"
[10] "ST.depression"        "Slope.of.ST"          "Number.of.vessels.fluro"
[13] "Thallium"              "Heart.Disease"
> # Convert categorical variable to factor
> data$Sex <- as.factor(data$Sex)
> # Apply One-way ANOVA
> anova_result <- aov(Cholesterol ~ Sex, data = data)
> # Display ANOVA table
> summary(anova_result)
summary(anova_result)
   Df Sum Sq Mean Sq F value    Pr(>F)
Sex      1  29220  29220  11.36 0.000861 ***
Residuals 268 689404   2572
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

MVLU COLLEGE

Subject:-Data Analysis with SAS / SPSS /R

8 Performing two-way ANOVA using aov() (R).

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Shows R code for performing a two-way ANOVA on the Indian Weather Dataset.
- Console:** Displays the R session output, including the ANOVA table and significance codes.
- File Explorer:** Shows a list of files in the current project directory, including CSV files and R scripts.
- Environment:** Shows the global environment with various objects like `anova_result`, `boston`, etc.

```
1 # Practical 8: Two-Way ANOVA (Indian Weather Dataset)
2
3 # Load dataset
4 data <- read.csv("indian_weather_data.csv")
5
6 # Convert categorical variables to factors
7 data$city <- as.factor(data$city)
8 data$weather_code <- as.factor(data$weather_code)
9
10 # Apply Two-way ANOVA with interaction
11 anova_result <- aov(temperature ~ city * weather_code, data = data)
12
13 # Display ANOVA summary
14 summary(anova_result)
15 |
```

```
R> # Load dataset
R> data <- read.csv("indian_weather_data.csv")
R> # Convert categorical variables to factors
R> data$city <- as.factor(data$city)
R> data$weather_code <- as.factor(data$weather_code)
R> # Apply Two-way ANOVA with interaction
R> anova_result <- aov(temperature ~ city * weather_code, data = data)
R> # Display ANOVA summary
R> summary(anova_result)

   Df Sum Sq Mean Sq F value Pr(>F)
city    1    1788   25.18 9.754e+28 <2e-16 ***
Residuals  2      0  0.00

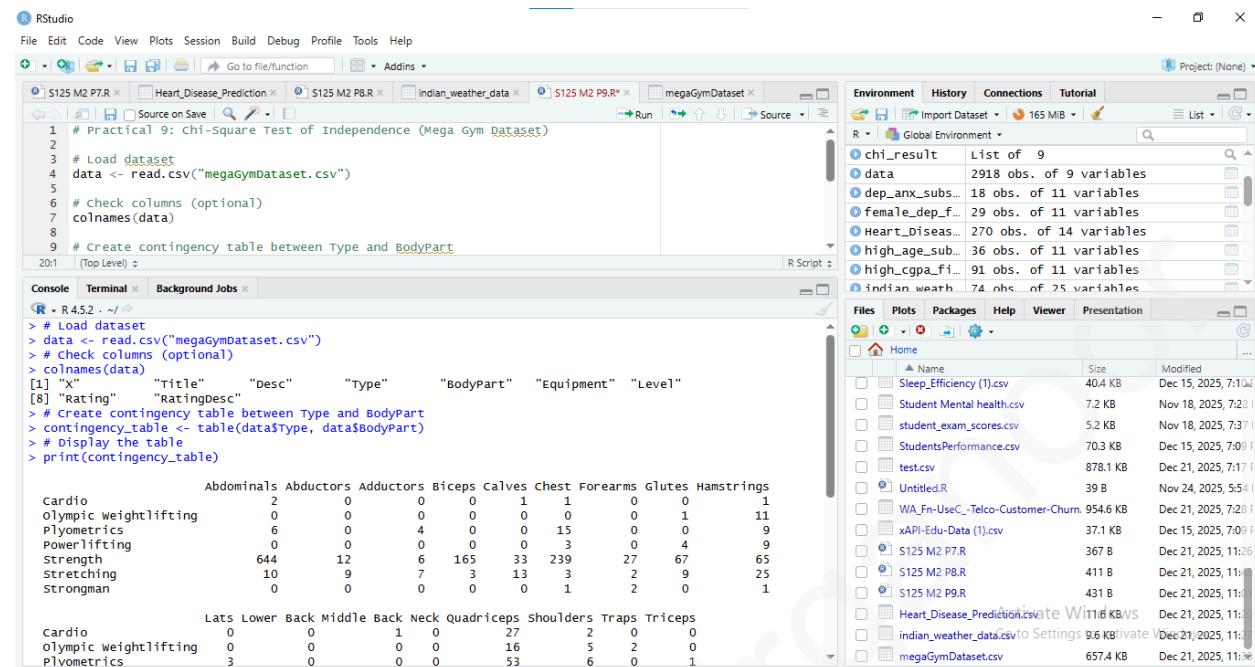
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Name	Size	Modified
Sleep_Efficiency (1).csv	40.4 KB	Dec 15, 2025, 7:10 AM
Student_Mental_health.csv	7.2 KB	Nov 18, 2025, 7:22 AM
student_exam_scores.csv	5.2 KB	Nov 18, 2025, 7:37 AM
StudentsPerformance.csv	70.3 KB	Dec 15, 2025, 7:09 PM
test.csv	878.1 KB	Dec 21, 2025, 7:17 PM
Untitled.R	39 B	Nov 24, 2025, 5:54 AM
WA_Fn-UseC_-Telco-Customer-Churn.csv	954.6 KB	Dec 21, 2025, 7:28 PM
xAPI-Edu-Data (1).csv	37.1 KB	Dec 15, 2025, 7:09 PM
S125 M2 P7.R	367 B	Dec 21, 2025, 11:26 PM
S125 M2 P8.R	409 B	Dec 21, 2025, 11:27 PM
S125 M2 P9.R	431 B	Dec 21, 2025, 11:28 PM
Heart_Disease_Prediction.csv	11.6 KB	Dec 21, 2025, 11:29 PM
indian_weather_data.csv	9.6 KB	Dec 21, 2025, 11:29 PM
megaGymDataset.csv	657.4 KB	Dec 21, 2025, 11:29 PM

MVLU COLLEGE

Subject:-Data Analysis with SAS / SPSS /R

9 Conducting Chi-square tests using chisq.test() (R).



The screenshot shows the RStudio interface with the following code in the script pane:

```

1 # Practical 9: Chi-Square Test of Independence (Mega Gym dataset)
2
3 # Load dataset
4 data <- read.csv("megaGymDataset.csv")
5
6 # Check columns (optional)
7 colnames(data)
8
9 # Create contingency table between Type and BodyPart
20:1 (Top Level) +

```

The console output shows the creation of a contingency table:

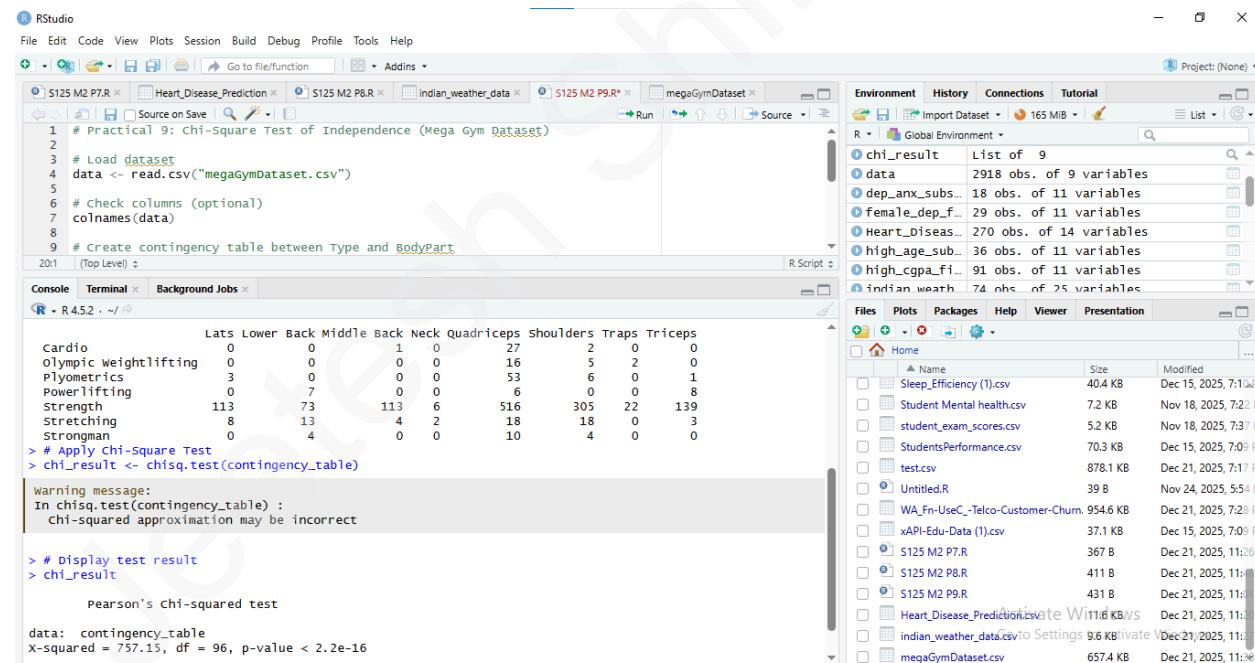
```

> # Load dataset
> data <- read.csv("megaGymDataset.csv")
> # Check columns (optional)
> colnames(data)
[1] "X"      "Title"   "Desc"    "Type"    "BodyPart" "Equipment" "Level"
[8] "Rating" 
> # Create contingency table between Type and BodyPart
> contingency_table <- table(data$type, data$bodyPart)
> # Display the table
> print(contingency_table)

          Abdominals Abdutors Adductors Biceps Calves Chest Forearms Glutes Hamstrings
Cardio           2       0       0     0     1     1       0       0       1
Olympic weightlifting   0       0       0     0     0     0       0       1      11
Plyometrics        6       0       4     0     0     15       0       0       9
Powerlifting        0       0       0     0     0     3       0       4       9
Strength          644      12      6    165    33    239      27      67      65
Stretching         10       9       7     3     13     3       2       9      25
Strongman          0       0       0     0     0     1       2       0       1

          Lats Lower Back Middle Back Neck quadriceps Shoulders Traps Triceps
Cardio           0       0       1     0     27      2       0       0       0
Olympic weightlifting   0       0       0     0     16      5      2       0       0
Plyometrics        3       0       0       0     53      6       0       1       1
Powerlifting        0       7       0     0     6       0       0       8       8
Strength          113      73     113     6    516    305      22     139
Stretching         8       13      4     2     18      18      0       3       3
Strongman          0       4       0     0     10      4       0       0       0

```



The screenshot shows the RStudio interface with the following code in the script pane:

```

1 # Practical 9: Chi-Square Test of Independence (Mega Gym dataset)
2
3 # Load dataset
4 data <- read.csv("megaGymDataset.csv")
5
6 # Check columns (optional)
7 colnames(data)
8
9 # Create contingency table between Type and BodyPart
20:1 (Top Level) +

```

The console output shows the application of chisq.test() and the resulting Pearson's Chi-squared test output:

```

> # Apply Chi-Square Test
> chi_result <- chisq.test(contingency_table)

Warning message:
In chisq.test(contingency_table) :
  Chi-squared approximation may be incorrect

> # Display test result
> chi_result

Pearson's Chi-squared test

data: contingency_table
X-squared = 757.19, df = 96, p-value < 2.2e-16

```