RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Project: (None)

Source

Console Terminal Background Jobs

R 4.4.2 · ~/

```
> 
> # Sample dataset
> train_data <- data.frame(Height = c(150, 160, 170, 180, 175), Weight = c(50, 6
0, 70, 80, 75))
> train_labels <- c("Underweight", "Normal", "Overweight", "Obese", "Overweight")
> 
> # Test data
> test_data <- data.frame(Height = c(165, 185), Weight = c(65, 85))
> 
> # Apply KNN
> predicted_labels <- knn(train_data, test_data, train_labels, k = 3)
> print(predicted_labels)
[1] Overweight Overweight
Levels: Normal Obese Overweight Underweight
> 
> # Sample dataset
> set.seed(123)
> data <- data.frame(X = rnorm(10, mean = 5, sd = 2), Y = rnorm(10, mean = 5, sd
= 2))
> 
> # Apply K-Means Clustering
> clusters <- kmeans(data, centers = 2)
> 
> # Print cluster results
> print(clusters$cluster)
 [1] 2 2 2 2 1 2 2 1 2 1
> 
> # Plot Clusters
> plot(data, col = clusters$cluster, pch = 19, main = "K-Means Clustering")
> 
> 
```

Environment History Connections Tutorial

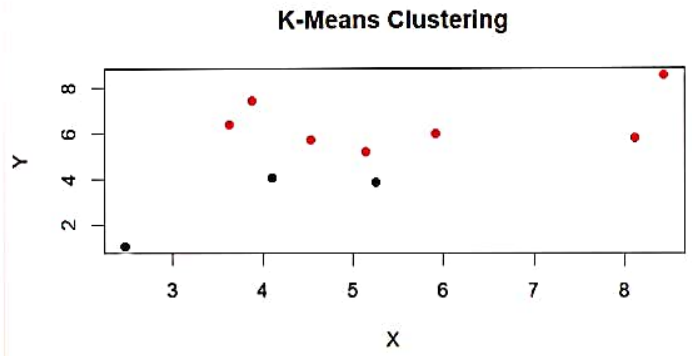
Import Dataset · 140 MiB · List

R · Global Environment

data	10 obs. of 2 variables
df_combined_colu…	3 obs. of 4 variables
df_combined_rows	4 obs. of 4 variables
df1	3 obs. of 2 variables
df2	3 obs. of 2 variables
df3	1 obs. of 4 variables
faithful	272 obs. of 2 variables

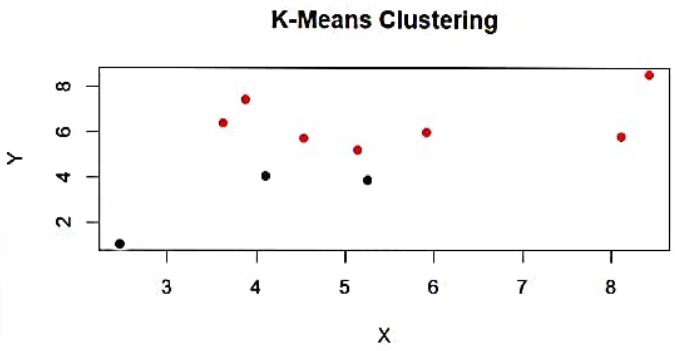
Files Plots Packages Help Viewer Presentation

Zoom Export · Publish



```
> 
> 
> 
> # 5x4 matrix filled by rows
> matrix1 <- matrix(1:20, nrow = 5, ncol = 4, byrow = TRUE)
> print(matrix1)
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
[5,]   17   18   19   20
> 
> # 3x3 matrix with labels
> matrix2 <- matrix(1:9, nrow = 3, ncol = 3, byrow = TRUE, dimnames = list(c("Row
1", "Row2", "Row3"), c("Col1", "Col2", "Col3")))
> print(matrix2)
     Col1 Col2 Col3
Row1    1    2    3
Row2    4    5    6
Row3    7    8    9
> 
> # 2x2 matrix filled by columns
> matrix3 <- matrix(1:4, nrow = 2, ncol = 2, byrow = FALSE, dimnames = list(c
("A", "B"), c("X", "Y")))
> print(matrix3)
  X Y
A 1 3
B 2 4
> 
> 
```

Environment	
matrix3	int [1:2, 1:2] 1 2 3 4
model	List of 12
test_data	2 obs. of 2 variables
train_data	5 obs. of 2 variables
Values	
avg_A	22.4285714285714
avg_B	19.2857142857143



File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Source

Console | Terminal | Background Jobs

R 4.4.2 · ~/

```
1  1   Alice  25      F
2  2     Bob  30      M
3  3 Charlie  22      M
4  4   David  28      M
>
>
>
> str <- "Hello, R Programming!"
>
> # Convert to uppercase
> print(toupper(str))
[1] "HELLO, R PROGRAMMING!"
>
> # Convert to lowercase
> print(tolower(str))
[1] "hello, r programming!"
>
> # Find substring
> print(substring(str, 8, 17))
[1] "R Programm"
>
> # Replace substring
> print(gsub("R", "Python", str))
[1] "Hello, Python Programming!"
>
> # Concatenate strings
> str2 <- "Let's Learn."
> print(paste(str, str2))
[1] "Hello, R Programming! Let's Learn."
>
>
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Source

Console Terminal × Background Jobs ×

R 4.4.2 · ~/

```
Coefficients:
             Estimate Std. Error    t value Pr(>|t|)
(Intercept) -5.000e+00  2.405e-15 -2.079e+15   <2e-16 ***
x            1.000e+00  7.252e-17  1.379e+16   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.293e-15 on 3 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 1.901e+32 on 1 and 3 DF,  p-value: < 2.2e-16

Warning message:
In summary.lm(model) : essentially perfect fit: summary may be unreliable
>
> library(class)
>
> # Sample dataset
> train_data <- data.frame(Height = c(150, 160, 170, 180, 175), Weight = c(50, 6
0, 70, 80, 75))
> train_labels <- c("Underweight", "Normal", "Overweight", "Obese", "Overweight")
>
> # Test data
> test_data <- data.frame(Height = c(165, 185), Weight = c(65, 85))
>
> # Apply KNN
> predicted_labels <- knn(train_data, test_data, train_labels, k = 3)
> print(predicted_labels)
[1] Overweight Overweight
Levels: Normal Obese Overweight Underweight
>
>
```

Source

Console Terminal × Background Jobs ×

R · R 4.4.2 · ~/

```
> # Create array
> arr_3d <- array(c(vector1, vector2), dim = c(3, 3, 2))
>
> # Print the array
> print(arr_3d)
, , 1

     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

, , 2

     [,1] [,2] [,3]
[1,]   10   13   16
[2,]   11   14   17
[3,]   12   15   18

>
>
>
> # Empty plot
> plot.new()
>
> # Empty plot with specified axes limits
> plot(1, type = "n", xlim = c(0, 10), ylim = c(0, 10), xlab = "X-axis", ylab =
"Y-axis", main = "Empty Plot with Limits")
>
>
```

Environment History Connections Tutorial

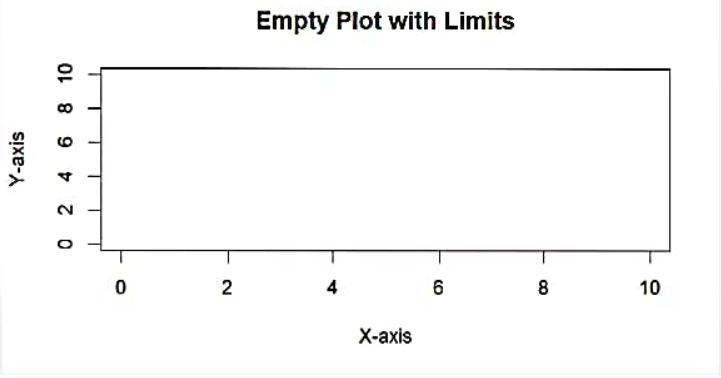
Import Dataset · 146 MiB · List

R · Global Environment

train_labels	chr [1:5] "Underweight" "Normal" "Overweight" "...
values	int [1:12] 1 2 3 4 5 6 7 8 9 10 ...
var_A	2.95238095238095
var_B	1.23809523809524
var_C	1.80952380952381
vector1	int [1:9] 1 2 3 4 5 6 7 8 9
vector2	int [1:9] 10 11 12 13 14 15 16 17 18

Files Plots Packages Help Viewer Presentation

Zoom · Export · Publish



File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

Source

Console | Terminal | Background Jobs

R 4.4.2 · ~/

```
Row2    2    5    8   11
Row3    3    6    9   12

>
>
>
> # Define vectors
> vector1 <- c(1:9)
> vector2 <- c(10:18)
>
> # Create array
> arr_3d <- array(c(vector1, vector2), dim = c(3, 3, 2))
>
> # Print the array
> print(arr_3d)
, , 1

     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

, , 2

     [,1] [,2] [,3]
[1,]   10   13   16
[2,]   11   14   17
[3,]   12   15   18

>
> |
```

Environment | History | Connections | Tutorial

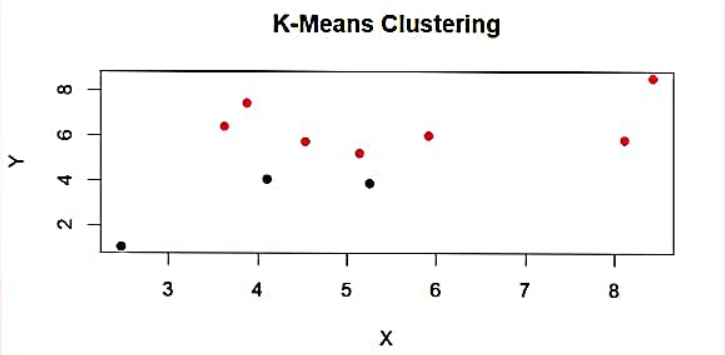
Import Dataset | 145 MiB | List

R | Global Environment

train_labels	chr [1:5] "Underweight" "Normal" "Overweight" "…
values	int [1:12] 1 2 3 4 5 6 7 8 9 10 ...
var_A	2.95238095238095
var_B	1.23809523809524
var_C	1.80952380952381
vector1	int [1:9] 1 2 3 4 5 6 7 8 9
vector2	int [1:9] 10 11 12 13 14 15 16 17 18

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish



File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Source

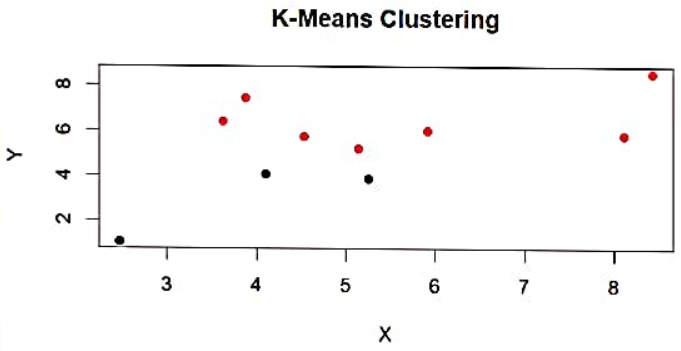
Console Terminal × Background Jobs ×

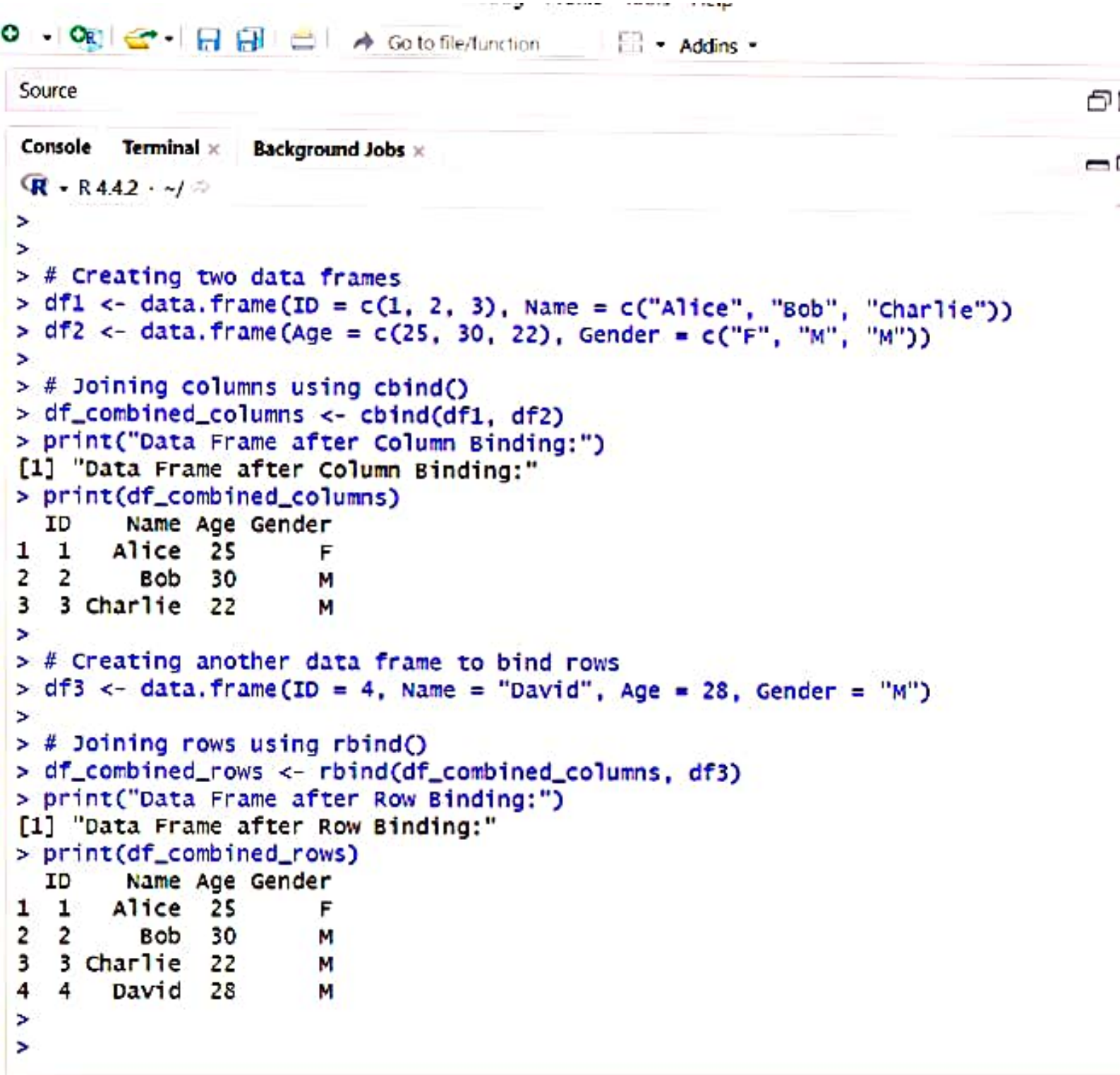
R 4.4.2 · ~/

```
> 
> # Sample data
> x <- c(10, 20, 30, 40, 50)
> y <- c(5, 15, 25, 35, 45)
> 
> # Calculate Correlation
> correlation <- cor(x, y)
> print(paste("Correlation:", correlation))
[1] "Correlation: 1"
> 
> # Calculate Covariance
> covariance <- cov(x, y)
> print(paste("Covariance:", covariance))
[1] "Covariance: 250"
> 
> # Linear Regression Model
> model <- lm(y ~ x)
> summary(model)

Call:
lm(formula = y ~ x)

Residuals:
         1          2          3          4          5 
 3.657e-16 -2.481e-15  2.342e-15  1.297e-15 -1.524e-15 

Coefficients:
             Estimate Std. Error    t value Pr(>|t|)    
(Intercept) -5.000e+00  2.405e-15 -2.079e+15   <2e-16 ***
x            1.000e+00  7.252e-17  1.379e+16   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> # Print cluster results
> print(clusters$cluster)
 [1] 2 2 2 2 1 2 2 1 2 1
>
> # Plot Clusters
> plot(data, col = clusters$cluster, pch = 19, main = "K-Means Clustering")
>
>
> # Numeric vector
> num_vector <- c(10, 20, 30, 40)
> print(num_vector)
[1] 10 20 30 40
> print(typeof(num_vector))
[1] "double"
>
> # Character vector
> char_vector <- c("Apple", "Banana", "Cherry")
> print(char_vector)
[1] "Apple"  "Banana" "Cherry"
> print(typeof(char_vector))
[1] "character"
>
> # Logical vector
> log_vector <- c(TRUE, FALSE, TRUE, FALSE)
> print(log_vector)
[1]  TRUE FALSE  TRUE FALSE
> print(typeof(log_vector))
[1] "logical"
>
>
```

highest_avg_city	"C"
highest_avg_value	31.8571428571429
log_vector	logi [1:4] TRUE FALSE TRUE FALSE
max_temp_A	25
max_temp_B	21
max_temp_C	34
num_vector	num [1:4] 10 20 30 40





Source

Console Terminal × Background Jobs ×

R 4.4.2 · ~/

```
>
>
> # Creating two data frames
> df1 <- data.frame(ID = c(1, 2, 3), Name = c("Alice", "Bob", "Charlie"))
> df2 <- data.frame(Age = c(25, 30, 22), Gender = c("F", "M", "M"))
>
> # Joining columns using cbind()
> df_combined_columns <- cbind(df1, df2)
> print("Data Frame after Column Binding:")
[1] "Data Frame after Column Binding:"
> print(df_combined_columns)
  ID    Name Age Gender
1  1   Alice  25      F
2  2     Bob  30      M
3  3 Charlie  22      M
>
> # Creating another data frame to bind rows
> df3 <- data.frame(ID = 4, Name = "David", Age = 28, Gender = "M")
>
> # Joining rows using rbind()
> df_combined_rows <- rbind(df_combined_columns, df3)
> print("Data Frame after Row Binding:")
[1] "Data Frame after Row Binding:"
> print(df_combined_rows)
  ID    Name Age Gender
1  1   Alice  25      F
2  2     Bob  30      M
3  3 Charlie  22      M
4  4   David  28      M
>
>
```

```
Row1    1    2    3
Row2    4    5    6
Row3    7    8    9
>
> # 2x2 matrix filled by columns
> matrix3 <- matrix(1:4, nrow = 2, ncol = 2, byrow = FALSE, dimnames = list(c
("A", "B"), c("X", "Y")))
> print(matrix3)
  X Y
A 1 3
B 2 4
>
>
>
> # Define values and dimensions
> values <- c(1:12)
> dimensions <- c(3, 4, 1)
>
> # Create array with named dimensions
> arr <- array(values, dim = dimensions, dimnames = list(c("Row1", "Row2", "Row
3"), c("Col1", "Col2", "Col3", "Col4"), "Table1"))
> print(arr)
, , Table1

     Col1 Col2 Col3 Col4
Row1    1    4    7   10
Row2    2    5    8   11
Row3    3    6    9   12

>
>
```

max_temp_C	34
num_vector	num [1:4] 10 20 30 40
predicted_labels	Factor w/ 4 levels "Normal","Obese",..: 3 3
str	"Hello, R Programming!"
str2	"Let's Learn."
train_labels	chr [1:5] "Underweight" "Normal" "Overweight" "...
values	int [1:12] 1 2 3 4 5 6 7 8 9 10 ...