

ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

Create a new R program - my x +

mycompiler.io/new/r

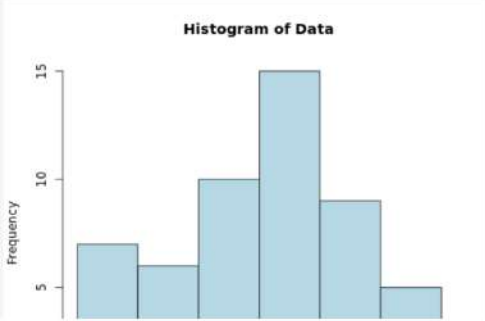
RunSave

```
1
2 data <- c(1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15,
3          18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25,
4          25, 25, 25, 28, 28, 30, 30, 30)
5
6
7 n <- length(data)
8 bins <- split(data, cut(seq_along(data), 3, labels = FALSE))
9
10
11 bin_means <- lapply(bins, mean)
12 data_smooth_means <- unlist(mapply(function(b, m) rep(m, length(b)), bins, bin_means))
13
14
15 data_smooth_boundaries <- unlist(lapply(bins, function(b) {
16   min_b <- min(b)
17   max_b <- max(b)
18   b[b < (min_b + max_b) / 2] <- min_b
19   b[b >= (min_b + max_b) / 2] <- max_b
20   return(b)
21 })))
22
23 # Step 3: Plot Histogram
24 hist(data, breaks=10, col="lightblue", main="Histogram of Data", xlab="Value", border="black")
25
```

Program input


Output

[Execution complete with exit code 0]



A histogram titled "Histogram of Data" showing the frequency distribution of the data. The x-axis is labeled "Value" and the y-axis is labeled "Frequency". The histogram has 10 bins, with the highest frequency bin (the 5th bin) reaching a frequency of 15. The bars are light blue with black borders.

Bin	Frequency
1	7
2	6
3	10
4	15
5	9
6	5
7	0
8	0
9	0
10	0




Build your website for just \$3.88/mth. More value and performance with Namecheap.

RDS VIA CARBON

Supported languages

DenoJavaScriptNode.JSPythonRubyGo



ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

Create a new R program - my X

mycompiler.io/new/r

RunSave

```
1
2 ages <- c(13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33,
3         35, 35, 35, 35, 36, 40, 45, 46, 52, 70)
4
5
6 summary(ages)
7
8
9 sd(ages)
10 var(ages)
11
12
13 hist(ages, main="Age Distribution", xlab="Age", col="lightblue", border="black")
14
15
16 boxplot(ages, main="Boxplot of Ages", col="orange", horizontal=TRUE)
17 |
```

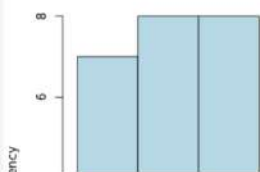
Program input


Output

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	13.00	20.50	25.00	29.96	35.00	70.00
[1]	12.94212					
[1]	167.4986					

[Execution complete with exit code 0]

Age Distribution






Build your website for just \$3.88/mth. More value and performance with Namecheap.

RDS VIA CARBON

Supported languages

DenoJavaScriptNode.JSPythonRubyGo



ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

Create a new R program - my X +

mycompiler.io/new/r

Wed 26 Feb 10:31 PM

R

RunSave

```
1
2 data <- matrix(c(15, 10, 5, 20, 25, 15, 10, 30, 20), nrow = 3, byrow = TRUE)
3
4
5 rownames(data) <- c("Age_Group_1", "Age_Group_2", "Age_Group_3")
6 colnames(data) <- c("Photo_A", "Photo_B", "Photo_C")
7
8
9 test_result <- chisq.test(data)
10
11
12 print(test_result)
13
```


Program input

Output

Pearson's Chi-squared test

data: data
X-squared = 11.311, df = 4, p-value = 0.02328

[Execution complete with exit code 0]




Build your website for just \$3.88/mth. More value and performance with Namecheap.

RDS VIA CARBON

Supported languages

DenoJavaScriptNode.JSPythonRubyGo



Chrome

File Edit View History Bookmarks Profiles Tab Window Help

Wed 26 Feb 10:34 PM

Create a new R program - my

mycompiler.io/new/r

R

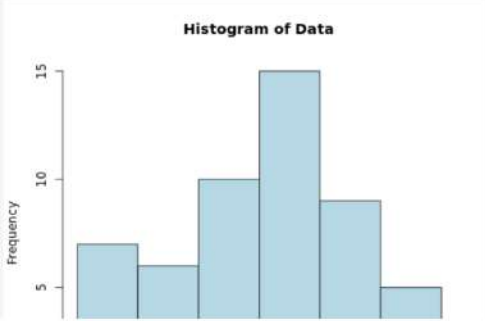
Run Save

```
1
2 class_A <- c(76, 35, 47, 64, 95, 66, 89, 36, 84)
3 class_B <- c(51, 56, 84, 60, 59, 70, 63, 66, 50)
4
5
6 mean_A <- mean(class_A)
7 median_A <- median(class_A)
8 range_A <- range(class_A)
9 range_A_value <- max(class_A) - min(class_A)
10
11 mean_B <- mean(class_B)
12 median_B <- median(class_B)
13 range_B <- range(class_B)
14 range_B_value <- max(class_B) - min(class_B)
15
16
17 cat("Class A - Mean:", mean_A, "Median:", median_A, "Range:", range_A_value, "\n")
18 cat("Class B - Mean:", mean_B, "Median:", median_B, "Range:", range_B_value, "\n")
19
20 boxplot(class_A, class_B, names = c("Class A", "Class B"),
21         col = c("lightblue", "lightgreen"), main = "Comparison of Class Scores",
22         ylab = "Scores", border = "black")
23
24
```

Program input


Output

[Execution complete with exit code 0]



A histogram titled "Histogram of Data" showing the frequency distribution of scores. The x-axis represents scores and the y-axis represents frequency. The bars are light blue. The frequencies for the score bins are approximately: 76-84: 7, 35-47: 6, 47-64: 10, 64-84: 15, 84-95: 9, 95-100: 5.

Score Range	Frequency
76-84	7
35-47	6
47-64	10
64-84	15
84-95	9
95-100	5




Build your website for just \$3.88/mth. More value and performance with Namecheap.

RDS VIA CARBON

Supported languages

Deno JavaScript NodeJS Python Ruby Go



Chrome

File Edit View History Bookmarks Profiles Tab Window Help

mycompiler.io/new/r

Run Save

1

2 data <- AirPassengers

3 hist(data,

4 breaks = seq(100, 700, by = 150), # Bins starting at 100, width = 150

5 col = "lightblue",

6 main = "Histogram of AirPassengers Dataset",

7 xlab = "Number of Passengers",

8 border = "black")

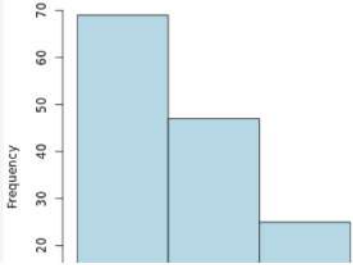
9

Program input


Output

[Execution complete with exit code 0]

Histogram of AirPassengers Dataset



Number of Passengers (Bin)	Frequency
100 - 250	68
250 - 400	48
400 - 550	25




Build your website for just \$3.88/mth. More value and performance with Namecheap.

RDS VIA CARBON

Supported languages

Deno JavaScript NodeJS Python Ruby Go



ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

Create a new R program - my X +

mycompiler.io/new/r

RunSave


```
1
2 data <- c(200, 300, 400, 600, 1000)
3 min_val <- min(data)
4 max_val <- max(data)
5 min_max_norm <- (data - min_val) / (max_val - min_val)
6
7
8 mean_val <- mean(data)
9 sd_val <- sd(data)
10 z_score_norm <- (data - mean_val) / sd_val
11
12 # Print results
13 cat("Min-Max Normalized Data:\n", min_max_norm, "\n")
14 cat("Z-Score Normalized Data:\n", z_score_norm, "\n")
15
```

Program input

Output


Min-Max Normalized Data:
0 0.125 0.25 0.5 1
Z-Score Normalized Data:
-0.9486833 -0.6324555 -0.3162278 0.3162278 1.581139

[Execution complete with exit code 0]

 Build your website for just \$3.88/mth. More value and performance with Namecheap.
RDS VIA CARBON

Supported languages

DenoJavaScriptNode.JSPythonRubyGo



ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

Create a new R program - my x +

mycompiler.io/new/r

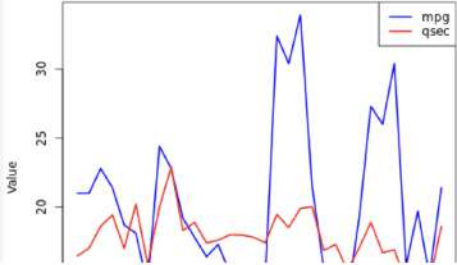
RunSave

```
1
2 data <- mtcars
3
4
5 plot(data$mpg, type = "l", col = "blue", lwd = 2, ylim = range(c(data$mpg, data$qsec)),
6      xlab = "Index", ylab = "Value", main = "Multiple Line Plot of mpg and qsec")
7
8 lines(data$qsec, col = "red", lwd = 2)
9 legend("topright", legend = c("mpg", "qsec"), col = c("blue", "red"), lwd = 2)
10
```


Program input

Output

[Execution complete with exit code 0]



A line plot titled "Multiple Line Plot of mpg and qsec". The x-axis is labeled "Index" and ranges from 1 to 32. The y-axis is labeled "Value" and ranges from 15 to 35. There are two data series: "mpg" (blue line) and "qsec" (red line). The "mpg" series starts at approximately 21, peaks at 26, drops to 15, peaks again at 33, and ends at 21. The "qsec" series starts at approximately 16, peaks at 20, drops to 15, peaks again at 20, and ends at 16.




Build your website for just \$3.88/mth. More value and performance with Namecheap.

RDS VIA CARBON

Supported languages

DenoJavaScriptNode.JSPythonRubyGo



ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

Create a new R program - my x +

mycompiler.io/new/r

RunSave

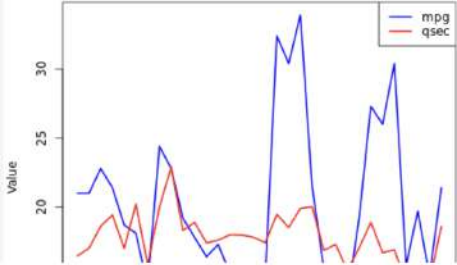
```
1
2 data <- mtcars
3 plot(data$mpg, type = "l", col = "blue", lwd = 2, ylim = range(c(data$mpg, data$qsec)),
4      xlab = "Index", ylab = "Value", main = "Multiple Line Plot of mpg and qsec")
5
6 lines(data$qsec, col = "red", lwd = 2)
7 legend("topright", legend = c("mpg", "qsec"), col = c("blue", "red"), lwd = 2)
8
```


Program input

Output

[Execution complete with exit code 0]

Multiple Line Plot of mpg and qsec






Build your website for just \$3.88/mth. More value and performance with Namecheap.

RDS VIA CARBON

Supported languages

DenoJavaScriptNode.JSPythonRubyGo



ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

Create a new R program - my X +

mycompiler.io/new/r

RunSave

```
1
2 scores <- c(15, 18, 25, 30, 22, 27, 35, 40, 100, 20) # 100 is a possible outlier
3 boxplot(scores,
4         col = "lightblue",
5         main = "Boxplot of Tennis Players' Scores",
6         ylab = "Points Scored",
7         horizontal = FALSE,
8         notch = TRUE)
9
10 grid()
11
```

Program input

Output

Warning message:
In (function (z, notch = FALSE, width = NULL, varwidth
some notches went outside hinges ('box'): maybe set r

[Execution complete with exit code 0]

Boxplot of Tennis Players' Scores

A boxplot titled "Boxplot of Tennis Players' Scores" showing the distribution of points scored. The y-axis is labeled "ts Scored" and ranges from 60 to 100. The boxplot shows a median around 25, with a box from approximately 18 to 35. There is a single outlier at 100.

Statistic	Value
Minimum	15
Q1	18
Median	25
Q3	35
Maximum	100

Build your website for just \$3.88/mth. More value and performance with Namecheap.

RDS VIA CARBON

Supported languages

DenoJavaScriptNode.JSPythonRubyGo

ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

Create a new R program - my x +

mycompiler.io/new/r

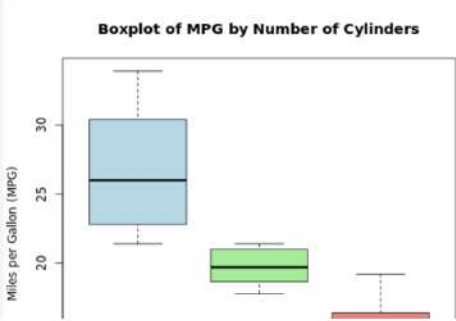
RunSave

```
1
2 data <- mtcars
3 boxplot(mpg ~ cyl, data = data,
4         col = c("lightblue", "lightgreen", "lightcoral"),
5         main = "Boxplot of MPG by Number of Cylinders",
6         xlab = "Number of Cylinders",
7         ylab = "Miles per Gallon (MPG)")
8
```

Program input

Output


[Execution complete with exit code 0]



Boxplot of MPG by Number of Cylinders

The boxplot displays the distribution of Miles per Gallon (MPG) for three cylinder categories: 4, 6, and 8. The y-axis represents MPG, ranging from 20 to 30. The 4-cylinder group (light blue) has the highest median MPG (around 26.5), followed by the 6-cylinder group (light green, median around 19.5), and the 8-cylinder group (light coral, median around 15.5). The plot shows that as the number of cylinders increases, the median MPG decreases.

Cylinders	Median MPG	Q1 MPG	Q3 MPG	Min MPG	Max MPG
4	26.5	23.0	30.0	21.5	33.0
6	19.5	18.0	20.5	17.0	22.0
8	15.5	14.5	16.5	13.0	19.0



Build your website for just \$3.88/mth. More value and performance with Namecheap.

RDS VIA CARBON

Supported languages

DenoJavaScriptNode.JSPythonRubyGo

