

Figure 1.

 **You**

write racket code that finds the mean, range, minimum, maximum and % of students that passed an exam give only a list of results. The % required to pass an exam is 30%

 **ChatGPT**

Certainly! Below is an example of Racket code that calculates the mean, range, minimum, maximum, and the percentage of students who passed an exam, given a list of results:

```
racket Copy code

(define (exam-analysis results)
  (define total-students (length results))

  ; Calculate mean
  (define mean (/ (apply + results) total-students))

  ; Calculate range
  (define range (- (apply max results) (apply min results)))

  ; Calculate minimum and maximum
  (define minimum (apply min results))
  (define maximum (apply max results))

  ; Calculate percentage of students who passed (scored >= 30%)
  (define pass-threshold (* 0.3 (apply + results)))
  (define passed-count (count (lambda (score) (>= score pass-threshold)) results))
  (define pass-percentage (/ (* 100 passed-count) total-students))

  ; Return the analysis results
  (list mean range minimum maximum pass-percentage))

; Example usage
(define exam-results '(60 75 90 45 30 80))
(define analysis-results (exam-analysis exam-results))

; Display the results
(display "Mean: ")
(display (first analysis-results))
(newline)

(display "Range: ")
(display (second analysis-results))
(newline)

(display "Minimum: ")
(display (third analysis-results))
(newline)

(display "Maximum: ")
(display (fourth analysis-results))
(newline)

(display "Pass Percentage: ")
(display (fifth analysis-results))
(display "%")
(newline)
```

Replace the `exam-results` list with the actual results of the students. The code calculates the mean, range, minimum, maximum, and the percentage of students who passed the exam based on a passing threshold of 30%.

Figure 2.

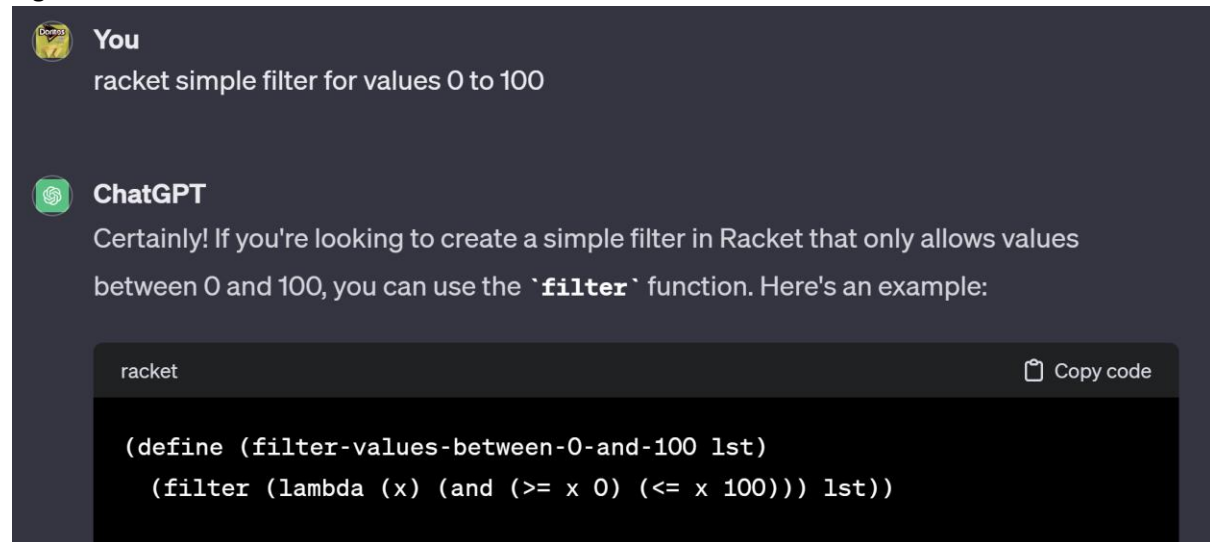


Figure 3.

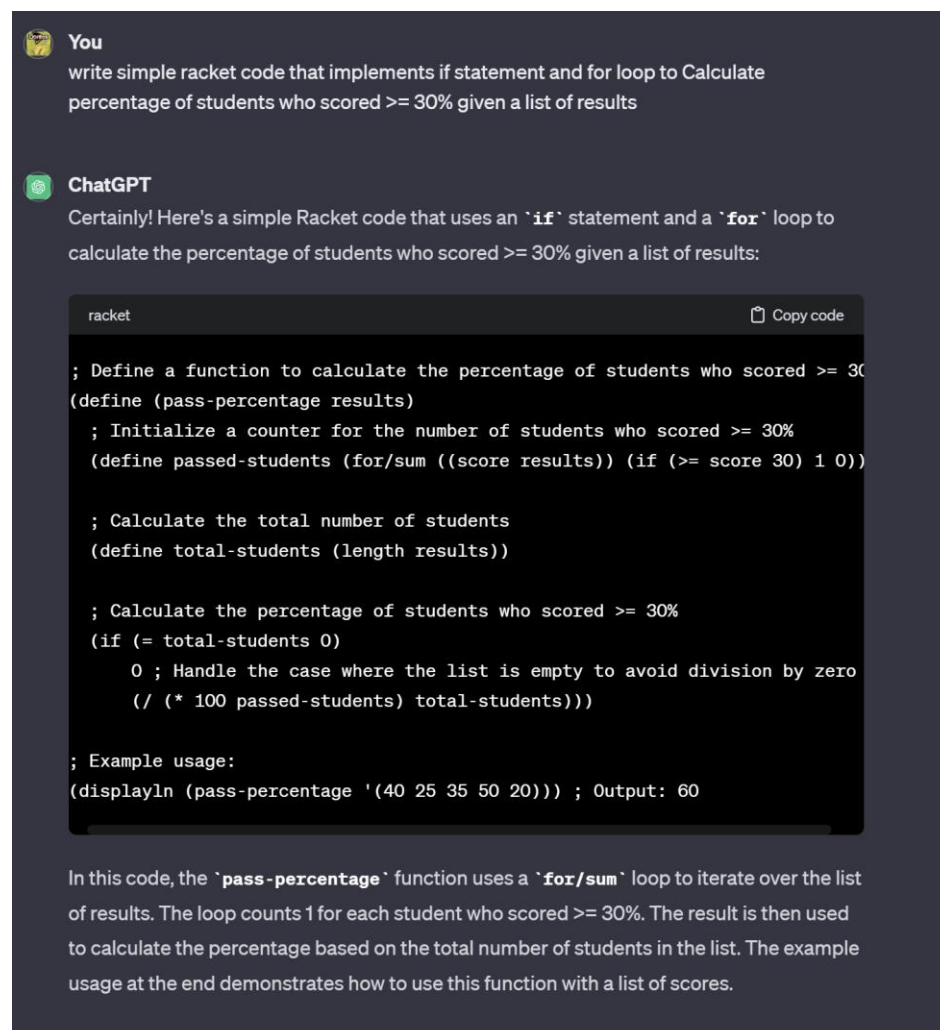




Figure 4.

 **You**  
how to i display fractions as decimals in racket

 **ChatGPT**  
In Racket, you can display fractions as decimals using the `exact->inexact` function to convert the fractions to inexact numbers, and then use the `display` or `printf` functions to show them as decimals. Here's an example:

racket Copy code

```
(define fraction (/ 3 4)) ; Replace with your own fraction

(define decimal (exact->inexact fraction))

(display "Fraction: ")
(display fraction)
(newline)

(display "Decimal: ")
(display decimal)
(newline)
```

In this example, the `fraction` variable contains the fraction `3/4`. The `exact->inexact` function is then used to convert it to a decimal (inexact number). Finally, the `display` function is used to print both the original fraction and its decimal representation.

You can replace the `fraction` variable with your own fraction, and the code will display both the fraction and its decimal equivalent.