Lab Report 6

Joshua Tran

2023-04-25

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

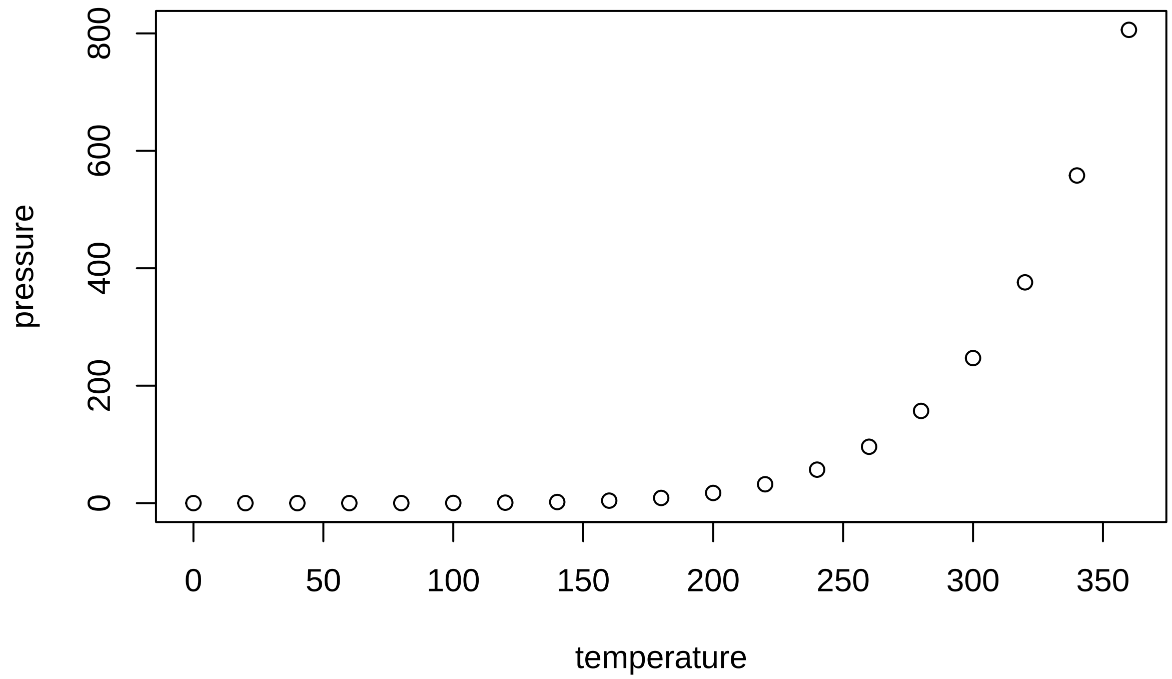
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
#Exampleinputvectorstostartwith
student1<-c(100,100,100,100,100,100,100,90)
student2<-c(100,NA,90,90,90,90,97,80)
student3<-c(90,NA,NA,NA,NA,NA,NA,NA)
```

```
# Load required packages
# Load required packages
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Dataset
student1 <- c(100, 100, 100, 100, 100, 100, 100, 90)
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)

# Combine dataset into a data frame
gradebook <- data.frame(Student = c("Student1", "Student2", "Student3"),
                         Assignment1 = c(student1[1], student2[1], student3[1]),
                         Assignment2 = c(student1[2], student2[2], student3[2]),
                         Assignment3 = c(student1[3], student2[3], student3[3]),
                         Assignment4 = c(student1[4], student2[4], student3[4]),
                         Assignment5 = c(student1[5], student2[5], student3[5]),
                         Assignment6 = c(student1[6], student2[6], student3[6]),
                         Assignment7 = c(student1[7], student2[7], student3[7]),
                         Assignment8 = c(student1[8], student2[8], student3[8]))

# Function to calculate the overall grade
calculate_grade <- function(scores) {
  # Replace NAs with the lowest possible score (0)
  scores[is.na(scores)] <- 0

  # Remove the lowest score
  scores <- scores[-which.min(scores)]

  # Calculate the average of remaining scores
  mean(scores)
}

# Apply the function to each student and add the results to the gradebook
gradebook <- gradebook %>%
  rowwise() %>%
  mutate(OverallGrade = calculate_grade(c(Assignment1, Assignment2, Assignment3, Assignment4, Assignmen

# Print the updated gradebook with overall grades
print(gradebook)
```

```
## # A tibble: 3 x 10
## # Rowwise:
##   Student  Assignment1 Assignment2 Assignment3 Assignment4 Assignment5
##   <chr>          <dbl>       <dbl>       <dbl>       <dbl>       <dbl>
## 1 Student1         100         100         100         100         100
## 2 Student2         100          NA          90          90          90
## 3 Student3          90          NA          NA          NA          NA
## # i 4 more variables: Assignment6 <dbl>, Assignment7 <dbl>, Assignment8 <dbl>,
## #   OverallGrade <dbl>
```

```r
url <- 'https://tinyurl.com/gradeinput'

classdata <- read.csv(url, row.names = 1)
head(classdata)
```

```
##           hw1 hw2 hw3 hw4 hw5
## student-1 100  73 100  88  79
## student-2  85  64  78  89  78
## student-3  83  69  77 100  77
## student-4  88  NA  73 100  76
## student-5  88 100  75  86  79
## student-6  89  78 100  89  77
```

```
# Function to calculate the overall grade
calculate_grade <- function(scores) {
  # Replace NAs with the lowest possible score (0)
  scores[is.na(scores)] <- 0

  # Remove the lowest score
  scores <- scores[-which.min(scores)]

  # Calculate the average of remaining scores
  mean(scores)
  #return(scores)
}
```

```
apply(classdata, 1, calculate_grade)
```

```
##  student-1  student-2  student-3  student-4  student-5  student-6  student-7
##      91.75      82.50      84.25      84.25      88.25      89.00      94.00
##  student-8  student-9 student-10 student-11 student-12 student-13 student-14
##      93.75      87.75      79.00      86.00      91.75      92.25      87.75
## student-15 student-16 student-17 student-18 student-19 student-20
##      78.75      89.50      88.00      94.50      82.75      82.75
```

Q1:

```
# Function to calculate the overall grade
calculate_grade <- function(scores) {
  # Replace NAs with the lowest possible score (0)
  scores[is.na(scores)] <- 0

  # Remove the lowest score
  scores <- scores[-which.min(scores)]

  # Calculate the average of remaining scores
  mean(scores)
  #return(scores)
}

apply(classdata, 1, calculate_grade)
```

```
##  student-1  student-2  student-3  student-4  student-5  student-6  student-7
##      91.75      82.50      84.25      84.25      88.25      89.00      94.00
##  student-8  student-9 student-10 student-11 student-12 student-13 student-14
##      93.75      87.75      79.00      86.00      91.75      92.25      87.75
## student-15 student-16 student-17 student-18 student-19 student-20
##      78.75      89.50      88.00      94.50      82.75      82.75
```

Q2:

```
#Sets the vector for all the student's final grades
student_scores <- apply(classdata, 1, calculate_grade)
# Determine the top-scoring student
top_student <- student_scores[which.max(student_scores)]

# Print the top-scoring student's name and grade
top_student
```

```
## student-18
##       94.5
```

Q3: NAs were excluded from the calculations

```
#Replace NAs with 0'z
all_student_grade_zeroes <- replace(classdata, is.na(classdata), 0)

# Calculate the mean score for each homework assignment
mean_scores <- colMeans(all_student_grade_zeroes)

# Find the toughest homework
toughest_homework <- names(which.min(mean_scores))

#print out toughest_homework without having set NAs to 0
toughest_homework
```

```
## [1] "hw2"
```

Q4:

```
#Using student_scores from Q2 to determine correlation
correlations <- apply(all_student_grade_zeroes, 2, cor, y=student_scores)

#Whats my max?
cor_index <- which.max(correlations)
correlations[cor_index]
```

```
##       hw5
## 0.6325982
```