MovieData_R_Assignment

2023-06-11

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.

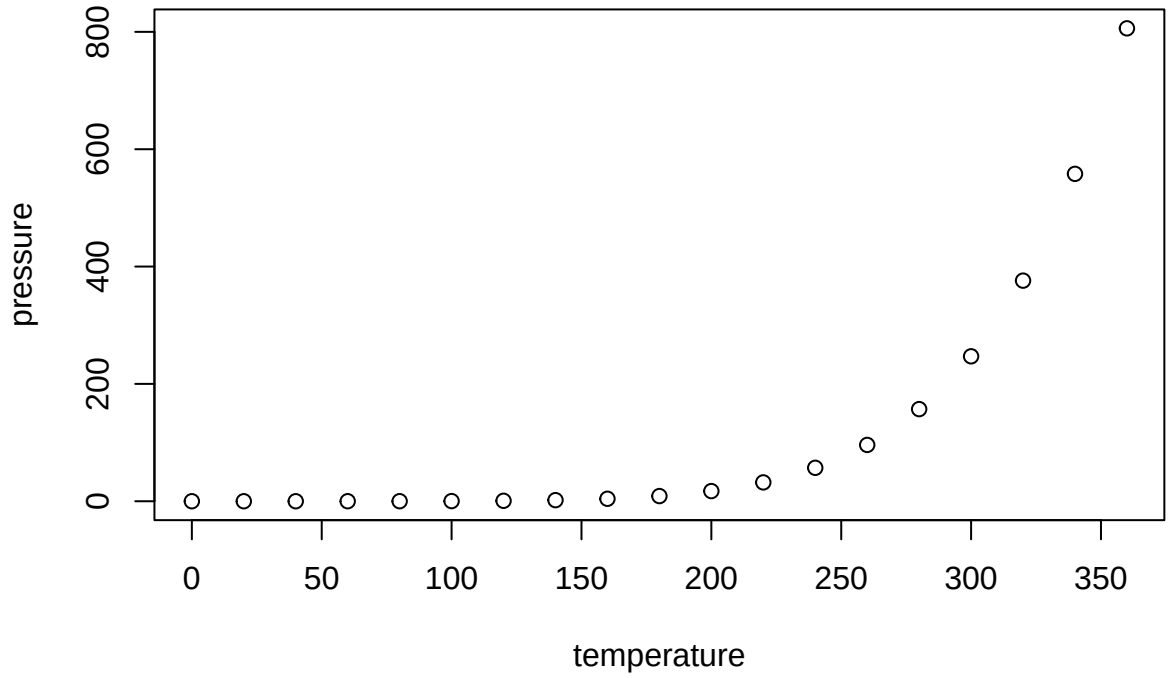
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
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.

my_log <- file("C:/Users/golak/Downloads/Assignment1_output.txt")

sink("C:/Users/golak/Downloads/Assignment1_output.txt")

Loading Dataset

library("readxl") dataset = read_excel("C:/Users/golak/Downloads/Movie Data.xlsx") View(dataset)

Print the structure of your dataset

str(dataset)

List the variables in your dataset

names(dataset)

Print the top 15 rows of your dataset

head(dataset, 15)

Write a user defined function using any of the variables from the data set.

my_function <- function(dataset, Rating) { variable <- dataset[[Rating]] average <- mean(variable) return(average) } my_function result <- my_function(dataset, "Rating") result

Use data manipulation techniques and filter rows based on any logical criteria that exist in your dataset

library(dplyr) filtered_data <- filter(dataset, Rating > 5) filtered_data

Identify the dependent & independent variables and use reshaping techniques and create a new data frame by joining those variables from your dataset.

Remove missing values in your dataset.

cleaned_data <- na.omit(dataset) cleaned_data <- na.omit(dataset) cleaned_data

Identify and remove duplicated data in your dataset

duplicated_rows <- duplicated(dataset) duplicated_rows

Reorder multiple rows in descending order

ordered_dataset <- arrange(dataset, desc(Year), desc(Rating)) ordered_dataset

Rename some of the column names in your dataset

colnames(dataset) names(dataset)[2]<-"release_year" names(dataset)[8]<-"movie_rating" colnames(dataset)

Add new variables in your data frame by using a mathematical function (for e.g. – multiply an existing column by 2 and add it as a new variable to your data frame)

dataset$New_Rating <- -dataset$movie_rating + 1 colnames(dataset)

Create a training set using random number generator engine.

training_indices <- sample(nrow(dataset), size = round(0.7 * nrow(dataset))) training_set <- dataset[training_indices,] training_set <- dataset[training_indices,] training_set

Print the summary statistics of your dataset

summary(dataset)

Use any of the numerical variables from the dataset and perform the following statistical functions • Mean • Median • Mode • Range

mean_rating <- mean(dataset$movie_rating) print(mean_rating)

median_rating <- median(dataset$movie_rating) print(median_rating)

calculate_mode <- function(x) { unique_values <- unique(x) counts <- tabulate(match(x, unique_values)) mode <- unique_values[which.max(counts)] return(mode) } variable_mode <- calculate_mode(variable) variable_mode

variable_range <- range(variable) variable_range

Plot a scatter plot for any 2 variables in your dataset

x <- dataset$movie_rating$y <- -dataset$release_year plot(x,y)

Plot a bar plot for any 2 variables in your dataset

variable <- dataset$movie_rating barplot(variable)

Find the correlation between any 2 variables by applying least square linear regression model

x <- dataset$movie_ratingy < −dataset$release_year correlation <- cor(x,y) correlation

sink()