Chad Huntebrinker’s Homework 4

Chad Huntebrinker

2024-09-22

Question 1:

Load the R built-in dataset “longley”.

Perform a summary analysis on the “GNP.deflator” and “Employed” variables in the “longley” dataset.

Identify the correlation between “GNP.deflator” and “Employed” in the subset of records prior to 1950.

Calculate a new variable “Product” by multiplying “GNP.deflator” and “Employed” for each record.

Calculate the average value of the “Product” variable.

Print only the records in the “longley” dataset prior to 1950, including the new “Product” variable.

Use R “head()” function to print the first 7 rows of the “longley” dataset.

Export the resulting subset of the “longley” dataset to a tab-delimited text file named “longley\_subset.txt”.

#Question 1:  
  
#Load the R built-in dataset "longley".  
data("longley")  
  
#Perform a summary analysis on the "GNP.deflator" and "Employed" variables in the "longley" dataset.  
summary(longley$GNP.deflator)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 83.00 94.53 100.60 101.68 111.25 116.90

summary(longley$Employed)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 60.17 62.71 65.50 65.32 68.29 70.55

#Identify the correlation between "GNP.deflator" and "Employed" in the subset of records prior to 1950.  
data\_prior\_1950 <- subset(longley, longley$Year < 1950)  
correlation <- cor(data\_prior\_1950$GNP.deflator, data\_prior\_1950$Employed)  
print(correlation)

## [1] 0.410317

#Calculate a new variable "Product" by multiplying "GNP.deflator" and "Employed" for each record.  
Product <- longley$GNP.deflator \* longley$Employed  
  
#Calculate the average value of the "Product" variable.  
mean(Product)

## [1] 6676.011

#Print only the records in the "longley" dataset prior to 1950, including the new "Product" variable.  
longley$Product <- Product  
product\_prior\_1950 <- subset(longley, longley$Year < 1950)  
print(product\_prior\_1950)

## GNP.deflator GNP Unemployed Armed.Forces Population Year Employed  
## 1947 83.0 234.289 235.6 159.0 107.608 1947 60.323  
## 1948 88.5 259.426 232.5 145.6 108.632 1948 61.122  
## 1949 88.2 258.054 368.2 161.6 109.773 1949 60.171  
## Product  
## 1947 5006.809  
## 1948 5409.297  
## 1949 5307.082

#Use R "head()" function to print the first 7 rows of the "longley" dataset.  
head(longley, 7)

## GNP.deflator GNP Unemployed Armed.Forces Population Year Employed  
## 1947 83.0 234.289 235.6 159.0 107.608 1947 60.323  
## 1948 88.5 259.426 232.5 145.6 108.632 1948 61.122  
## 1949 88.2 258.054 368.2 161.6 109.773 1949 60.171  
## 1950 89.5 284.599 335.1 165.0 110.929 1950 61.187  
## 1951 96.2 328.975 209.9 309.9 112.075 1951 63.221  
## 1952 98.1 346.999 193.2 359.4 113.270 1952 63.639  
## 1953 99.0 365.385 187.0 354.7 115.094 1953 64.989  
## Product  
## 1947 5006.809  
## 1948 5409.297  
## 1949 5307.082  
## 1950 5476.236  
## 1951 6081.860  
## 1952 6242.986  
## 1953 6433.911

#Export the resulting subset of the "longley" dataset to a tab-delimited text file named "longley\_subset.txt".  
write.table(head(longley, 7), file = "longley\_subset.txt")

Question 2:

Load the R built-in dataset “airquality “

Assign the rownames as AQrecord1, ACrecord2 and so on for all the rows of airquality

What is the mean of “Ozone” and “Solar.R” columns?

Check if NA values are existing. Use the complete.cases function to identify the NA values and separate the airquality into two different dataframes one with record containing NA values (airquality.NA) and the second without NA (airquality.WNA) values. How many records are in airquality.NA and without airquality.WNA?

Export the airquality.WNA as a “airquality.WNA.csv” file.

#Question 2:  
  
#Load the R built-in dataset “airquality “  
data("airquality")  
  
#Assign the rownames as AQrecord1, AQrecord2 and so on for all the rows of airquality  
rownames(airquality) <- paste("AQrecord", 1:nrow(airquality))  
  
#What is the mean of "Ozone" and "Solar.R" columns?  
mean(airquality$Ozone, na.rm = TRUE)

## [1] 42.12931

mean(airquality$Solar.R, na.rm = TRUE)

## [1] 185.9315

#Check if NA values are existing. Use the complete.cases function to identify the NA values and   
#separate the airquality into two different dataframes one with   
#record containing NA values (airquality.NA) and the second without NA (airquality.WNA) values.   
#How many records are in airquality.NA and without airquality.WNA?  
sum(is.na(airquality))

## [1] 44

airquality.NA <- subset(airquality, complete.cases(airquality) == FALSE)  
airquality.WNA <- subset(airquality, complete.cases(airquality))  
  
dimension\_of\_na <- dim(airquality.NA)  
dimension\_of\_wna <- dim(airquality.WNA)  
  
print(dimension\_of\_na[1])

## [1] 42

print(dimension\_of\_wna[1])

## [1] 111

#Export the airquality.WNA as a “airquality.WNA.csv” file.  
write.csv(airquality.WNA, file = "airquality.WNA.csv")

Question 3:

This question requires you to install and load “tidyverse” package before you start working.

You are provided with exp\_data.csv file that contains the data from an experiment.

Read this data into a dataframe “exp.data” and then convert this into long format into “data\_long” dataframe.

Export the data\_long as “exp\_data\_long.csv”

Convert the long format back into wide format into a dataframe “data\_wide”

#Question 3:  
  
#This question requires you to install and load “tidyverse” package before you start working.  
#You are provided with exp\_data.csv file that contains the data from an experiment.  
#Read this data into a dataframe “exp.data” and then convert this into long format into “data\_long” dataframe.  
#Export the data\_long as “exp\_data\_long.csv”  
#Convert the long format back into wide format into a dataframe “data\_wide”  
library(conflicted)  
library("tidyverse")

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2

exp.data <- read.csv("exp\_data.csv")  
  
# Convert to long format  
data\_long <- exp.data %>%  
 pivot\_longer(cols = !Sample\_ID,  
 names\_to = "OD",  
 values\_to = "Data")  
print(data\_long)

## # A tibble: 100 × 3  
## Sample\_ID OD Data  
## <chr> <chr> <dbl>  
## 1 Sample01 OD\_D1 0.968  
## 2 Sample01 OD\_D2 0.857  
## 3 Sample01 OD\_D3 0.921  
## 4 Sample01 OD\_D4 0.921  
## 5 Sample01 OD\_D5 0.827  
## 6 Sample01 OD\_D6 0.833  
## 7 Sample01 OD\_D7 0.812  
## 8 Sample01 OD\_D8 0.790  
## 9 Sample01 OD\_D9 0.768  
## 10 Sample01 OD\_D10 0.746  
## # ℹ 90 more rows

write.csv(data\_long, file = "exp\_data\_long.csv")  
data\_wide <- data\_long %>%   
 pivot\_wider(names\_from = OD,  
 values\_from = Data)

Question 4:

Load the DNase dataset. Display the first few rows of the dataset

Calculate the mean and standard deviation of the ‘density’ column.

Create a new column ‘conc\_category’ based on the ‘conc’ values (e.g., Low, Medium, High)

Calculate the correlation between ‘conc’ and ‘density’. Check the relation between concentration and density by generating a plot

Calculate the percentage change in ‘density’ from the first row to the last row

Export the modified dataframe as a table (e.g., CSV format)

#Load the DNase dataset. Display the first few rows of the dataset  
data("DNase")  
head(DNase)

## Run conc density  
## 1 1 0.04882812 0.017  
## 2 1 0.04882812 0.018  
## 3 1 0.19531250 0.121  
## 4 1 0.19531250 0.124  
## 5 1 0.39062500 0.206  
## 6 1 0.39062500 0.215

#Calculate the mean and standard deviation of the 'density' column.  
mean(DNase$density)

## [1] 0.7191591

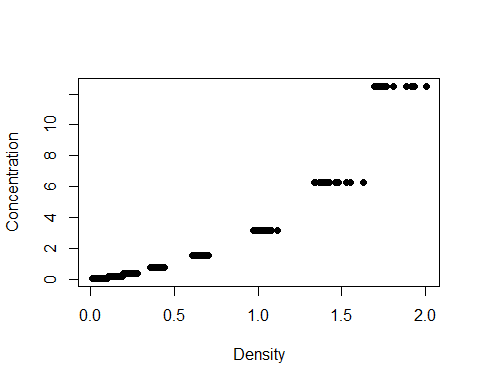
sd(DNase$density)

## [1] 0.5955726

#Create a new column 'conc\_category' based on the 'conc' values (e.g., Low, Medium, High)  
DNase$conc\_category <- ifelse(DNase$conc < 1, "Low",  
 ifelse(DNase$conc <= 5, "Medium", "High"))  
  
#Calculate the correlation between 'conc' and 'density'. Check the relation between concentration and density by generating a plot  
cor(DNase$density, DNase$conc)

## [1] 0.9309673

plot(DNase$density, DNase$conc, pch = 16, xlab = "Density", ylab = "Concentration")



#Calculate the percentage change in 'density' from the first row to the last row  
cat(((DNase$density[nrow(DNase)] - DNase$density[1]) / DNase$density[1]) \* 100, "%", sep = "")

## 10023.53%

#Export the modified dataframe as a table (e.g., CSV format)  
write.csv(DNase, file = "DNase\_file.csv")