Statistical Computing with R Assignment

Dagim Alemu

2025-02-19

#1 Generate a random sample of size 250, using the middle four digits of your ID in set.seed(), from chi-square distribution with 2 degrees of freedom. Provide descriptive summaries for your data and also justify the reason why chose those summaries. Draw histogram of your data and comment on it.

# Answer

# first lets set or generate the randome variable below

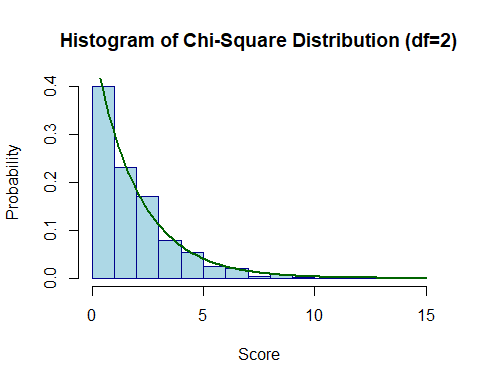
set.seed(9713) # my ID: GSR/9713/17 Dagim Alemu  
randNums <- rchisq(420, df = 2) # this generates 420 samples with a degree of freedom of 2 and setting them to a variable called randNums  
head(randNums)

# basic summary of the generated data above

library(e1071)  
summary(randNums) # maximum , min , quartile first and third second being median 🫡  
sd(randNums) # Standard deviation  
var(randNums) # Variance  
  
skewness(randNums) # it measures asymmetry meaning could be left or right skewed   
kurtosis(randNums) # it measures wherether or not the data is picked or not   
  
# Further justification and reasons Justification:Mean, Median, Min, Max: Give an overview of central tendency and range of data. SD, Variance: Measure the spread of the data. Skewness, Kurtosis: These help analyze the distribution's shape; skewness for asymmetry, kurtosis for tail weight.

# Histogram plot

# histogram plot  
hist(randNums, breaks = 20, col = "lightblue", border = "darkblue",  
 main = "Histogram of Chi-Square Distribution (df=2)",  
 xlab = "Score", ylab = "Probability", probability = TRUE)  
  
# Add Chi-Square probability curve for comparison  
curve(dchisq(x, df = 2), col = "darkgreen", lwd = 2, add = TRUE)



# Description: This histogram shows how often different scores appear   
# in our sample from a Chi-Square distribution with 2 degrees of freedom.   
# The blue bars represent our data, while the green line shows the expected   
# shape of this distribution.

# 2 second question

# loading stata fiile first

library(haven) # should install haven for package first to load dta for stata files first since the data is using the latest stata file version   
women15\_49 <- read\_dta('women15\_49.dta')  
head(women15\_49)

# setting with my middle id and taking 2500 rows from the data set

set.seed(9713)  
newDf <- women15\_49[sample(nrow(women15\_49), 2500), ]

# a) Provide socio-demographic(select five of them from the data) table by their knowledge where to get the test.

# converting v783 col to factor  
newDf$v783 <- factor(newDf$v783, levels = c(0, 1), labels = c("No Clue", "In the Know"))  
  
  
# v012:- Respondent's current age v024:- Region v106:- Highest educational level v130: Religion v151: Sex of household head  
  
# converting them to factors  
newDf$v024 <- as.factor(newDf$v024)  
newDf$v106 <- as.factor(newDf$v106)  
newDf$v130 <- as.factor(newDf$v130)  
newDf$v151 <- as.factor(newDf$v151)  
  
# create the table  
socio\_demographic\_table <- table(  
 Knows\_HIV\_Test = newDf$v783,  
 Age = newDf$v012,  
 Region = newDf$v024,  
 Education = newDf$v106,  
 Religion = newDf$v130,  
 Head\_Gender = newDf$v151  
)  
  
# Print the table  
print(socio\_demographic\_table)

1. Suggest the most appropriate model to assess association between your outcome vari able and set of your explanatory variables.

# I choose the logistic regression model because the outcome variable is binary (yes or no)  
# Using v012 (age), v024 (region), v106 (education), v130 (religion), and v151 (sex of household head) as explanatory variables  
  
# Fit logistic regression model  
model <- glm(v783 ~ v012 + v024 + v106 + v130 + v151,   
 data = newDf, family = binomial(link = "logit"))

# c) Using 10 fold cross-validation. Select the model that you think is the best.

# Logistic regression model with no special settings has RMSE of 0.463, R² is 0.088, and MAE is 0.423. This mean model explain little about knowing where to get HIV test. Because R² very low, we can make model much better.

# converted the dta file to csv i had multiple problems with the dta file so i was forced to work with csv file version of the file

# Load libraries  
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   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

# Load data and set seed  
data <- read.csv("women15\_49.csv", na.strings = "NA")  
set.seed(1234) # Replace with your ID's middle 4 digits  
sample\_data <- data %>% sample\_n(2500)  
  
# Clean outcome variable (v756: 1=Knows, 0=Doesn't know)  
sample\_data <- sample\_data %>%  
 mutate(know\_hiv\_test = ifelse(v756 == 1, 1, 0)) %>%  
 filter(!is.na(know\_hiv\_test))  
  
# Recode factors  
sample\_data <- sample\_data %>%  
 mutate(  
 age\_group = factor(v013, labels = c("15-19", "20-24", "25-29", "30-34", "35-39", "40-44", "45-49")),  
 education = factor(v106, labels = c("No Education", "Primary", "Secondary", "Higher")),  
 residence = factor(v025, labels = c("Urban", "Rural")),  
 region = factor(v024),  
 religion = factor(v130)  
 )  
  
  
#10-fold cross-validation  
ctrl <- trainControl(method = "cv", number = 10, savePredictions = TRUE)  
cv\_model <- train(  
 know\_hiv\_test ~ age\_group + education + residence + region + religion,  
 data = sample\_data,  
 method = "glm",  
 family = "binomial",  
 trControl = ctrl  
)

## Warning in train.default(x, y, weights = w, ...): You are trying to do  
## regression and your outcome only has two possible values Are you trying to do  
## classification? If so, use a 2 level factor as your outcome column.

# Output cross-validation results  
cv\_model$results

## parameter RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 1 none 0.4625751 0.0871813 0.4230237 0.01313914 0.02854083 0.01161499

# 3) Consider data on CD4 count for 50 HIV/AIDS patients.258, 268, 264, 286, 270, 231, 253, 220, 240, 230, 240, 260, 240, 296, 266, 227, 264, 264,230, 254, 262, 267, 241, 231, 237, 244, 269, 247, 224, 246, 244, 219, 253, 238, 244, 243, 258, 222, 246, 254, 253, 229, 247, 221, 255, 253, 248, 241, 215, 220 Using the middle four digits of your ID in set.seed(), select a random sample of size 30 to answer the following question. Using bootstrapping estimate mean and median number of CD4count of the patients, include 95% CI for your estimates.

# Asnwer

set.seed(9713)  
  
# CD4 counts of the patients   
cd4\_data <- c(258, 268, 264, 286, 270, 231, 253, 220, 240, 230, 240, 260, 240, 296, 266, 227, 264, 264,  
 230, 254, 262, 267, 241, 231, 237, 244, 269, 247, 224, 246, 244, 219, 253, 238, 244, 243,  
 258, 222, 246, 254, 253, 229, 247, 221, 255, 253, 248, 241, 215, 220)  
  
# Set sample size and iterations  
sample\_size <- 30  
iterations <- 10000 # number of bootstrap samples  
  
# func bootstrapping to calculate mean and median  
bootstrap\_results <- function(data, iterations, sample\_size) {  
 # empty vectors for storing results   
 mean\_values <- numeric(iterations)  
 median\_values <- numeric(iterations)  
   
 # bootstrapping  
 for (i in 1:iterations) {  
 CD4\_data\_sample <- sample(data, size = sample\_size, replace = TRUE)  
 mean\_values[i] <- mean(CD4\_data\_sample)  
 median\_values[i] <- median(CD4\_data\_sample)  
 }  
   
 # Calculate 95% confidence intervals  
 mean\_ci <- quantile(mean\_values, c(0.025, 0.975))  
 median\_ci <- quantile(median\_values, c(0.025, 0.975))  
   
 # results  
 return(list(mean = mean(mean\_values), mean\_ci = mean\_ci,  
 median = median(median\_values), median\_ci = median\_ci))  
}  
  
# Perform bootstrapping  
results <- bootstrap\_results(cd4\_data, iterations, sample\_size)  
  
# Output the results  
cat("Estimated Mean: ", results$mean, "\n")  
cat("95% CI for Mean: [", results$mean\_ci[1], ", ", results$mean\_ci[2], "]\n")  
cat("Estimated Median: ", results$median, "\n")  
cat("95% CI for Median: [", results$median\_ci[1], ", ", results$median\_ci[2], "]\n")

# The estimated mean CD4 count is 246.65, with a 95% confidence interval ranging from 240.47 to 253. The estimated median CD4 count is 246, with a 95% confidence interval between 240 and 254. This means we are 95% confident that the true mean and median CD4 counts fall within these intervals.