lab4.R

Feipeng Huang

2022-10-05

#Q1  
r\_mean = 10.4  
r\_sd = 2.4  
  
norm\_17 = rnorm(17, mean = r\_mean, sd = r\_sd)  
norm\_30 = rnorm(30, mean = r\_mean, sd = r\_sd)  
norm\_300 = rnorm(300, mean = r\_mean, sd = r\_sd)  
norm\_3000 = rnorm(3000, mean = r\_mean, sd = r\_sd)  
#Q2  
require(here)

## Loading required package: here

## here() starts at /Users/stonehuang/Documents/environmental\_data

png(  
 filename = here("lab\_04\_hist\_01.png"),  
 width = 1500, height = 1600,   
 res = 180, units = "px")  
  
par(mfrow = c(2, 2))  
  
hist(norm\_17, main = "Histogram 17 nd random numbers")  
hist(norm\_30, main = "Histogram 30 nd random numbers")  
hist(norm\_300, main = "Histogram 300 nd random numbers")  
hist(norm\_3000, main = "Histogram 3000 nd random numbers")  
  
dev.off()

## quartz\_off\_screen   
## 2

#Q4  
#Histograms of 17 and 30 normally-distributed random numbers are more skewed. Histograms of 300 and 3000 normally-distributed random numbers are less skewed and better resemble the shape of normal distribution.   
#Q5  
#The shapes of the histograms are different due to difference in sample size. A larger sample size can better represent the normal distribution.  
#Q6  
#mean = 0  
#standard deviation = 1  
#Q7  
x = seq(0, 20, length.out = 100)  
y = dnorm(x, mean = 10.4, sd = 2.4, log = FALSE)  
pdf(  
 file = here("norm\_1.pdf"),  
 width = 7, height = 7  
 )  
plot(x, y, xlim = c(0, 20), main = "Normal PDF: mean = 10.4, sd = 2.4", type = "l")  
abline(h = 0)  
dev.off()

## quartz\_off\_screen   
## 2

#Q9  
norm\_1 = rnorm(59, mean = 5.9, sd = 5.9)  
norm\_2 = rnorm(5959, mean = 5.9, sd = 5.9)  
unif\_1 = runif(n = 59, min = 5, max = 9)  
unif\_2 = runif(n = 5959, min = 5, max = 9)  
png(  
 filename = here("lab\_04\_Q10.png"),  
 width = 1500, height = 1600,   
 res = 180, units = "px")  
par(mfrow = c(2, 2))  
hist(norm\_1, main = "59 random normally-distributed numbers", col = rgb(1,1,0))  
hist(norm\_2, main = "5959 random normally-distributed numbers", col = rgb(1,0.5,0))  
hist(unif\_1, main = "59 random uniform numbers", col = rgb(1,1,0))  
hist(unif\_2, main = "5959 random uniform numbers", col = rgb(1,0.5,0))  
dev.off()

## quartz\_off\_screen   
## 2

#Q11  
# Calculates the value of y for a linear function, given the coordinates  
# of a known point (x1, y1) and the slope of the line.  
line\_point\_slope = function(x, x1, y1, slope)  
{  
 get\_y\_intercept =   
 function(x1, y1, slope)   
 return(-(x1 \* slope) + y1)  
   
 linear =   
 function(x, yint, slope)   
 return(yint + x \* slope)  
   
 return(linear(x, get\_y\_intercept(x1, y1, slope), slope))  
}  
library(here)  
n\_pts = 59  
x\_min = 5  
x\_max = 9  
x\_random = runif(n = n\_pts, min = x\_min, max = x\_max)  
y\_random = rnorm(n = n\_pts, mean = 5.9, sd = 5.9)  
dat\_random = data.frame(x = x\_random, y = y\_random)  
png(  
 filename = here("lab\_04\_Q12.png"),  
 width = 1500, height = 1600,   
 res = 180, units = "px")  
plot(y ~ x, data = dat\_random, pch = 8)  
guess\_x = 7  
guess\_y = 5.9  
guess\_slope = 0  
curve(line\_point\_slope(x, guess\_x, guess\_y, guess\_slope), add = T)  
dev.off()

## quartz\_off\_screen   
## 2

line\_point\_slope(dat\_random$x, guess\_x, guess\_y, guess\_slope)

## [1] 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9  
## [20] 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9  
## [39] 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9 5.9  
## [58] 5.9 5.9

#Q13

dat\_random$y\_predicted = line\_point\_slope(dat\_random$x, guess\_x, guess\_y, guess\_slope)  
dat\_random$resids = dat\_random$y - dat\_random$y\_predicted

#Q14

Chart, histogram

Description automatically generatedChart, box and whisker chart

Description automatically generated