coursework

Ehansa Gajanayake

2024-07-16

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(MASS)

##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:dplyr':  
##   
## select

Q1

First\_box <- c(1, 3, 5)  
Second\_box <- c(2, 6, 8)  
  
#a  
Possible\_Valuesof\_X <- c(  
 First\_box[1] + Second\_box[1], First\_box[1] + Second\_box[2], First\_box[1] + Second\_box[3],  
 First\_box[2] + Second\_box[1], First\_box[2] + Second\_box[2], First\_box[2] + Second\_box[3],  
 First\_box[3] + Second\_box[1], First\_box[3] + Second\_box[2], First\_box[3] + Second\_box[3]  
)  
  
Possible\_Valuesof\_X

## [1] 3 7 9 5 9 11 7 11 13

#b  
Frequencies <- table(Possible\_Valuesof\_X)  
  
Frequencies

## Possible\_Valuesof\_X  
## 3 5 7 9 11 13   
## 1 1 2 2 2 1

#Calculating PMFs for each value  
Pmf\_X\_3 <- Frequencies["3"] / sum(Frequencies)  
Pmf\_X\_5 <- Frequencies["5"] / sum(Frequencies)  
Pmf\_X\_7 <- Frequencies["7"] / sum(Frequencies)  
Pmf\_X\_9 <- Frequencies["9"] / sum(Frequencies)  
Pmf\_X\_11 <- Frequencies["11"] / sum(Frequencies)  
Pmf\_X\_13 <- Frequencies["13"] / sum(Frequencies)  
  
Pmf\_X\_3

## 3   
## 0.1111111

Pmf\_X\_5

## 5   
## 0.1111111

Pmf\_X\_7

## 7   
## 0.2222222

Pmf\_X\_9

## 9   
## 0.2222222

Pmf\_X\_11

## 11   
## 0.2222222

Pmf\_X\_13

## 13   
## 0.1111111

#c  
Expected\_Value\_of\_X <- (  
 (as.numeric(names(Frequencies)[1]) \* Pmf\_X\_3) +   
 (as.numeric(names(Frequencies)[2]) \* Pmf\_X\_5) +   
 (as.numeric(names(Frequencies)[3]) \* Pmf\_X\_7) +  
 (as.numeric(names(Frequencies)[4]) \* Pmf\_X\_9) +   
 (as.numeric(names(Frequencies)[5]) \* Pmf\_X\_11) +   
 (as.numeric(names(Frequencies)[6]) \* Pmf\_X\_13)  
)  
  
Expected\_Value\_of\_X

## 3   
## 8.333333

#Var(X) = E(X^2) - [E(X)]^2  
  
Expected\_Value\_of\_X2 <- (  
 (as.numeric(names(Frequencies)[1])^2 \* Pmf\_X\_3) +   
 (as.numeric(names(Frequencies)[2])^2 \* Pmf\_X\_5) +   
 (as.numeric(names(Frequencies)[3])^2 \* Pmf\_X\_7) +  
 (as.numeric(names(Frequencies)[4])^2 \* Pmf\_X\_9) +   
 (as.numeric(names(Frequencies)[5])^2 \* Pmf\_X\_11) +   
 (as.numeric(names(Frequencies)[6])^2 \* Pmf\_X\_13)  
)  
  
Expected\_Value\_of\_X2

## 3   
## 78.33333

Variance\_of\_X <- (Expected\_Value\_of\_X2 - ((Expected\_Value\_of\_X)^2))  
  
Variance\_of\_X

## 3   
## 8.888889

#d  
#Y=3X-4  
  
Poss\_Values\_Y = (3\*Possible\_Valuesof\_X)-4  
Poss\_Values\_Y

## [1] 5 17 23 11 23 29 17 29 35

Frequencies\_2 = table(Poss\_Values\_Y)  
Frequencies\_2

## Poss\_Values\_Y  
## 5 11 17 23 29 35   
## 1 1 2 2 2 1

Pmf\_X\_5 <- Frequencies\_2["5"] / sum(Frequencies\_2)  
Pmf\_X\_11 <- Frequencies\_2["11"] / sum(Frequencies\_2)  
Pmf\_X\_17 <- Frequencies\_2["17"] / sum(Frequencies\_2)  
Pmf\_X\_23 <- Frequencies\_2["23"] / sum(Frequencies\_2)  
Pmf\_X\_29 <- Frequencies\_2["29"] / sum(Frequencies\_2)  
Pmf\_X\_35 <- Frequencies\_2["35"] / sum(Frequencies\_2)  
  
Pmf\_X\_5

## 5   
## 0.1111111

Pmf\_X\_11

## 11   
## 0.1111111

Pmf\_X\_17

## 17   
## 0.2222222

Pmf\_X\_23

## 23   
## 0.2222222

Pmf\_X\_29

## 29   
## 0.2222222

Pmf\_X\_35

## 35   
## 0.1111111

#e  
cdf\_of\_Y1 = Pmf\_X\_5  
cdf\_of\_Y2 = cdf\_of\_Y1+Pmf\_X\_11  
cdf\_of\_Y3 = cdf\_of\_Y2+Pmf\_X\_17  
cdf\_of\_Y4 = cdf\_of\_Y3 + Pmf\_X\_23  
cdf\_of\_Y5 = cdf\_of\_Y4+Pmf\_X\_29  
cdf\_of\_Y = cdf\_of\_Y5+Pmf\_X\_35  
cdf\_of\_Y

## 5   
## 1

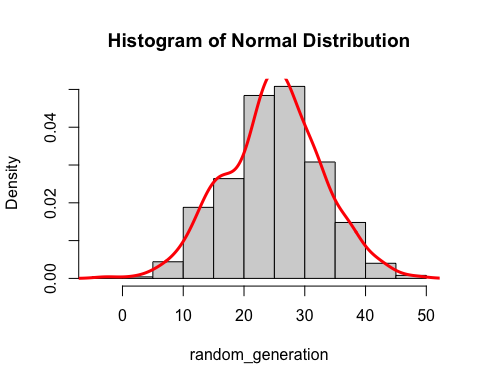
#f  
#P(Y=23)  
PY23 <- cdf\_of\_Y4 - cdf\_of\_Y3  
PY23

## 5   
## 0.2222222

Q2

number\_of\_values <- 500  
mean <- 25  
standard\_deviation <- 8  
random\_generation <- rnorm(number\_of\_values, mean, standard\_deviation)

#a  
hist(random\_generation, breaks=10, main="Histogram of Normal Distribution", freq=F)  
  
#b  
lines(density(random\_generation), lwd=3, col="red")

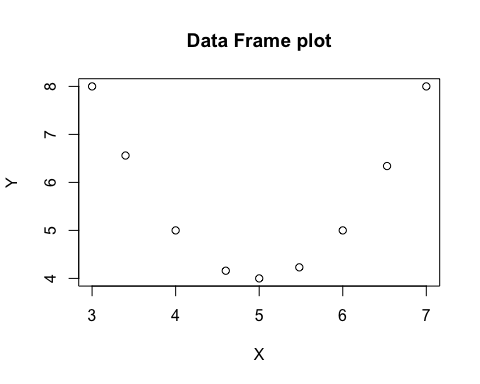


#c  
cat("We can see that the histogram and the density curve follow a normal distribution as the density curve is of a bell shape and the peak of the density curve is at the mean 25.")

## We can see that the histogram and the density curve follow a normal distribution as the density curve is of a bell shape and the peak of the density curve is at the mean 25.

Q3

#a  
X<- c(3.00, 3.40, 4.00, 4.60,5.00, 5.48, 6.00, 6.53, 7.00)  
Y <- c(8.0000, 6.5600, 5.0000, 4.1600, 4.0000, 4.2304, 5.0000, 6.3409, 8.0000)  
  
df <- data.frame(  
 X,Y  
)  
  
  
plot(df, main="Data Frame plot", xlab="X", ylab="Y")



#b  
correlation\_all\_data <- cor(X,Y)  
correlation\_all\_data

## [1] -0.02763248

#c   
cat("It is a very weak negative correlation, as the value is close to zero. This suggests that there is almost no linear relationship between X and Y.")

## It is a very weak negative correlation, as the value is close to zero. This suggests that there is almost no linear relationship between X and Y.

#d  
last\_six\_X <- X[4:9]  
last\_six\_Y <- Y[4:9]  
  
correlation\_last\_six <- cor(last\_six\_X, last\_six\_Y)  
correlation\_last\_six

## [1] 0.9206644

cat("This is a very strong positive correlation. This is because the the relationship between the last six pairs are more linear than the first few data. The first few data points shows a lot more variations.")

## This is a very strong positive correlation. This is because the the relationship between the last six pairs are more linear than the first few data. The first few data points shows a lot more variations.

#e  
X1 <- (3-(2\*X))  
X2 <- (2\*(X^3))  
  
correlation\_X1\_Y <- cor(X1, Y)  
correlation\_X2\_Y <- cor(X2, Y)  
  
correlation\_X1\_Y

## [1] 0.02763248

correlation\_X2\_Y

## [1] 0.1896496

cat("X1=3-2X is a linear transformation. rxy = -0.02763248 and rx1y = 0.02763248. Linear transformations do not change the nature of the correlation, but since X is multiplied by negative 2, the sign of the correlation coefficient is changed. \n")

## X1=3-2X is a linear transformation. rxy = -0.02763248 and rx1y = 0.02763248. Linear transformations do not change the nature of the correlation, but since X is multiplied by negative 2, the sign of the correlation coefficient is changed.

cat("X2 = 2X^3 is a non linear transformation. rxy = -0.02763248 and rx2y = 0.1896496. The non linear transformation has changed the relationship between the variables. The new correlation shows a weak positive linear relationship between X2 and Y as opposed to the negative almost zero correlation between X and Y.")

## X2 = 2X^3 is a non linear transformation. rxy = -0.02763248 and rx2y = 0.1896496. The non linear transformation has changed the relationship between the variables. The new correlation shows a weak positive linear relationship between X2 and Y as opposed to the negative almost zero correlation between X and Y.

Q4

cars %>%  
 view()

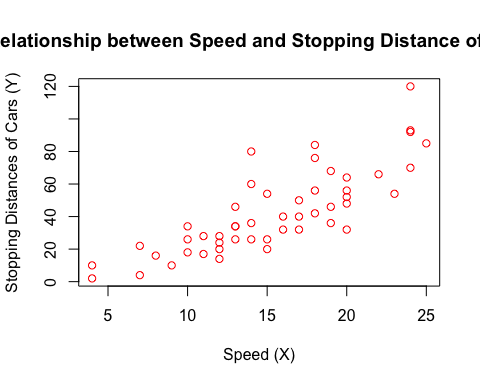
#a  
cars %>%   
 head(10)

## speed dist  
## 1 4 2  
## 2 4 10  
## 3 7 4  
## 4 7 22  
## 5 8 16  
## 6 9 10  
## 7 10 18  
## 8 10 26  
## 9 10 34  
## 10 11 17

#b  
print(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

#c  
plot(cars, xlab = "Speed (X)", ylab = "Stopping Distances of Cars (Y)",  
 main="Relationship between Speed and Stopping Distance of Cars",  
 col='red')



cat("It is a positive association where as the speed increases, the stopping distance of the cars also increase.")

## It is a positive association where as the speed increases, the stopping distance of the cars also increase.

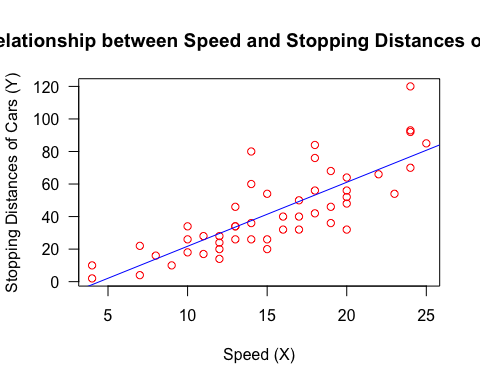
#d  
LRmodel <- lm(formula = dist ~ speed, data = cars) #linear model  
LRmodel

##   
## Call:  
## lm(formula = dist ~ speed, data = cars)  
##   
## Coefficients:  
## (Intercept) speed   
## -17.579 3.932

summary(LRmodel)

##   
## Call:  
## lm(formula = dist ~ speed, data = cars)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -29.069 -9.525 -2.272 9.215 43.201   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -17.5791 6.7584 -2.601 0.0123 \*   
## speed 3.9324 0.4155 9.464 1.49e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.38 on 48 degrees of freedom  
## Multiple R-squared: 0.6511, Adjusted R-squared: 0.6438   
## F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12

plot(cars, xlab = "Speed (X)", ylab = "Stopping Distances of Cars (Y)",las = 1,  
 main="Relationship between Speed and Stopping Distances of Cars",  
 col='red')  
abline (LRmodel,col ="blue")

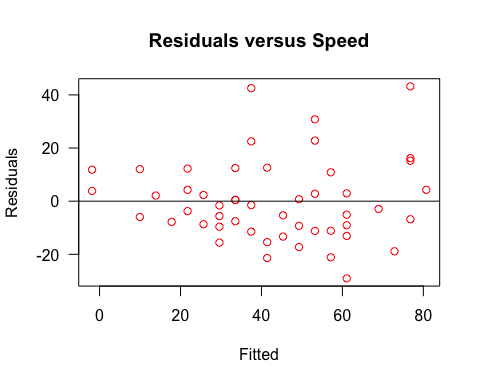


#e

#f  
cat("3.9324")

## 3.9324

#g  
plot(fitted(LRmodel),resid(LRmodel),  
 xlab = "Fitted", ylab ="Residuals",las = 1,   
 main="Residuals versus Speed",  
 col='red')   
abline(0,0)



cat("In this residual(errors) plot, the points do not seem to be randomly scattered around the residual=0 line. So we can conclude that   
a linear model is not appropriate for modeling this data. There are a few points with larger residuals, both positive and negative indicating possible outliers.")

## In this residual(errors) plot, the points do not seem to be randomly scattered around the residual=0 line. So we can conclude that   
## a linear model is not appropriate for modeling this data. There are a few points with larger residuals, both positive and negative indicating possible outliers.

#h  
new\_speed <- data.frame(speed = 21)  
predicted\_dist <- predict(LRmodel, newdata = new\_speed)  
predicted\_dist

## 1   
## 65.00149

equation\_predicted\_dist <- -17.5791+ 3.9324 \*21  
equation\_predicted\_dist

## [1] 65.0013

#Q5

#Part A

mean\_5 <- 4.11  
standard\_deviation\_5 = 1.37

#a  
answer\_0.85 <- qnorm(0.85, mean\_5, standard\_deviation\_5)  
answer\_0.85

## [1] 5.529914

#b  
answer\_0.35 = qnorm(0.35, mean\_5, standard\_deviation\_5)  
answer\_0.35

## [1] 3.582111

#c  
median <- mean\_5  
median

## [1] 4.11

#d  
morethan\_5dollar\_percentage <- pnorm(5, mean\_5, standard\_deviation\_5, lower.tail = F)\*100  
morethan\_5dollar\_percentage

## [1] 25.7964

Part B

#a  
cat("The most suitable distribution for X is binomial. This is because there is a fixed number of trials, 10 where a person can be either infected or not infected.The probability a person being infected in each trial is 0.1. \n")

## The most suitable distribution for X is binomial. This is because there is a fixed number of trials, 10 where a person can be either infected or not infected.The probability a person being infected in each trial is 0.1.

cat("n = 10 \n")

## n = 10

cat("p = 0.1 \n")

## p = 0.1

cat("X~Binomial(n,p) \n")

## X~Binomial(n,p)

cat("X~Binomial(10,0.1)")

## X~Binomial(10,0.1)

n\_times <- 10  
probability <- 0.1

#b  
less\_than\_3 = pbinom(2,n\_times,probability)  
less\_than\_3

## [1] 0.9298092

#c  
#E(X)=np  
#Var(X)=np(1-p)  
  
Mean\_of\_X = n\_times\*probability  
Variance\_of\_X2 = n\_times\*probability\*(1-probability)  
  
Mean\_of\_X

## [1] 1

Variance\_of\_X2

## [1] 0.9

#d  
cat("n = 100 \n")

## n = 100

cat("p = 0.024 \n")

## p = 0.024

cat("Since the number of trials n is very large and the probability of being infected in a single trial is very small, Poisson distribution can be used to approximate the binomial distribution. Set the Poisson mean to equal the binomial mean where, \n")

## Since the number of trials n is very large and the probability of being infected in a single trial is very small, Poisson distribution can be used to approximate the binomial distribution. Set the Poisson mean to equal the binomial mean where,

cat("λ = np = 100\*0.024 = 2.4 \n")

## λ = np = 100\*0.024 = 2.4

cat("So X ~ Pois(2.4)")

## So X ~ Pois(2.4)

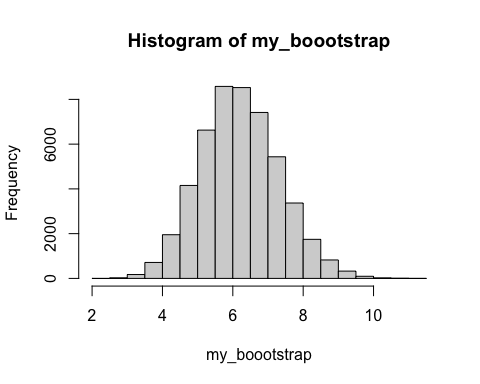
#Q6

N <- 50000 #number of bootstrap samples  
my\_boootstrap <- numeric(N) #vector where to store sample mean  
data <- c(1, 2, 3, 3, 5, 8, 7, 6, 5, 9, 11, 15) #given data set  
  
mean(data) #checking the mean of the given data

## [1] 6.25

for (i in 1:N) {  
 s=sample(data,12,replace = T)  
 my\_boootstrap[i]=mean(s)  
}

#a  
hist(my\_boootstrap)



#b  
quantile (my\_boootstrap, c(0.025, 0.975))

## 2.5% 97.5%   
## 4.166667 8.583333

#c  
qqnorm(my\_boootstrap)  
qqline(my\_boootstrap)

